Représentation et compression à haut niveau sémantique d'images 3D

Khouloud Samrouth

Thèse

THESE INSA Rennes
sous le sceau de l'Université européenne de Bretagne
pour obtenir le titre de
DOCTEUR DE L'INSA DE RENNES
Spécialité : Traitement du signal et d'image

présentée par
Khouloud Samrouth

ECOLE DOCTORALE : Matisse
LABORATOIRE : IETR

Représentation et compression à haut niveau sémantique d'images 3D

Thèse soutenue le 19.12.2014
devant le jury composé de :

Chaouki Diab
Professeur + ISSAE-CNAM-Liban / ***rapporteur et président***
Jean-Christophe BURIE
Professeur + Université la Rochelle / *rapporteur*
Vincent Ricordel
Maître de Conférences + Polytech'Nantes / *examinateur*
Luce Morin
Professeur + INSA de Rennes / *examinateur*
Bachar El Hassan
Maître de Conférences + Université Libanaise / *examinateur*
Wassim Falou
Maître de Conférences + Université Libanaise / *co-encadrant*
Mohamad Khalil
Professeur + Université Libanaise / Co-directeur de thèse
Olivier Deforges
Professeur + INSA de Rennes / Directeur de thèse

Représentation et compression à haut niveau sémantique d'images 3D

Khouloud Samrouth

En partenariat avec

Université Libanaise École Doctorale Sciences et Technologies Doyen				

L'Homme passionné par son travail
n'a pas le sentiment de travailler.

— Philippe Laurent, Formateur en entreprise

RÉSUMÉ

La diffusion de données multimédia, et particulièrement les images, continuent à croitre de manière très significative. La recherche de schémas de codage efficaces des images reste donc un domaine de recherche très dynamique. Aujourd'hui, une des technologies innovantes les plus marquantes dans ce secteur est sans doute le passage à un affichage 3D. La technologie 3D est largement utilisée dans les domaines de divertissement, d'imagerie médicale, de l'éducation et même plus récemment dans les enquêtes criminelles.

Il existe différentes manières de représenter l'information 3D. L'une des plus répandues consiste à associer à une image classique dite de texture, une image de profondeur de champs. Cette représentation conjointe permet ainsi une bonne reconstruction 3D dès lors que les deux images sont bien corrélées, et plus particulièrement sur les zones de contours de l'image de profondeur. En comparaison avec des images 2D classiques, la connaissance de la profondeur de champs pour les images 3D apporte donc une information sémantique importante quant à la composition de la scène.

Dans cette thèse, nous proposons ainsi un schéma de codage scalable d'images 3D de type 2D plus profondeur avec des fonctionnalités avancées, qui préserve toute la sémantique présente dans les images, tout en garantissant une efficacité de codage significative. La notion de préservation de la sémantique peut être traduite en termes de fonctionnalités telles que l'extraction automatique de zones d'intérêt, la capacité de coder plus finement des zones d'intérêt par rapport au fond, la recomposition de la scène et l'indexation.

Ainsi, dans un premier temps, nous introduisons un schéma de codage scalable et joint texture/profondeur. La texture est codée conjointement avec la profondeur à basse résolution, et une méthode de compression de la profondeur adaptée aux caractéristiques des cartes de profondeur est proposée.

Ensuite, nous présentons un schéma global de représentation fine et de codage basé contenu. Nous proposons ainsi un schéma global de représentation et de codage de "Profondeur d'Intérêt", appelé "Autofocus 3D". Il consiste à extraire finement des objets en respectant les contours dans la carte de profondeur, et de se focaliser automatiquement sur une zone de profondeur pour une meilleure qualité de synthèse.

Enfin, nous proposons un algorithme de segmentation en régions d'images 3D, fournissant une forte consistance entre la couleur, la profondeur et les régions de la scène. Basé sur une exploitation conjointe de l'information couleurs, et celle de profondeur, cet algorithme permet la segmentation de la scène avec un degré de granularité qui est fonction de l'application visée. Basé sur cette représentation en régions, il est possible d'appliquer simplement le même principe d'Autofocus 3D précédent, pour une extraction et un codage de la profondeur d'Intérêt (DoI).

L'élément le plus remarquable de ces deux approches est d'assurer une pleine cohérence spatiale entre texture, profondeur, et régions, se traduisant par une minimisation des problèmes de distorsions au niveau des contours et ainsi par une meilleure qualité dans les vues synthétisées.

ABSTRACT

Dissemination of multimedia data, in particular the images, continues to grow very significantly. Therefore, developing effective image coding schemes remains a very active research area. Today, one of the most innovative technologies in this area is the 3D technology. This 3D technology is widely used in many domains such as entertainment, medical imaging, education and very recently in criminal investigations.

There are different ways of representing 3D information. One of the most common representations, is to associate a depth image to a classic colour image called texture. This joint representation allows a good 3D reconstruction, as the two images are well correlated, especially along the contours of the depth image. Therefore, in comparison with conventional 2D images, knowledge of the depth of field for 3D images provides an important semantic information about the composition of the scene.

In this thesis, we propose a scalable 3D image coding scheme for 2D plus depth representation with advanced functionalities, which preserves all the semantics present in the images, while maintaining a significant coding efficiency. The concept of preserving the semantics can be translated in terms of features such as an automatic extraction of regions of interest, the ability to encode the regions of interest with higher quality than the background, a post-production of the scene and an indexing.

Thus, firstly we introduce a joint and scalable 2D plus depth coding scheme. First, texture is coded jointly with depth at low resolution, and a method of depth data compression well suited to the characteristics of the depth maps is proposed. This method exploits the strong correlation between the depth map and the texture to better encode the depth map.

Next, we present a global fine representation and content-based coding scheme. Therefore, we propose a representation and coding scheme based on "Depth of Interest", called "3D Autofocus". It consists in a fine extraction of objects, while preserving the contours in the depth map, and it allows to automatically focus on a particular depth zone, for a high rendering quality.

Finally, we propose a 3D image segmentation, providing a high consistency between colour, depth and regions of the scene. Based on a joint exploitation of the colour and depth information, this algorithm allows the segmentation of the scene with a level of granularity depending on the intended application. Based on such representation of the scene, it is possible to simply apply the same previous 3D Autofocus, for Depth of Interest extraction and coding.

It is remarkable that both approaches ensure a high spatial coherence between texture, depth, and regions, allowing to minimize the distortions along object of interest's contours and then a higher quality in the synthesized views.

REMERCIEMENTS

Feeling gratitude and
not expressing it is like
wrapping a present and
not giving it.

— William Arthur Ward

Mes premiers remerciements s'adressent à mes directeurs Pr. Olivier Deforges et Pr. Mohamad Khalil et à mon encadrant Dr. Wassim El Falou qui m'ont offert de l'aide et support durant ces trois années de thèse. Le travail avec mes trois encadrants a été une expérience très agréable de point de vue scientifique ainsi que de point de vue humain, notamment j'ai apprecie leur soutien lors de la fracture de mon épaule. Je vous remercie pour votre confiance dans mon travail. Mohamad et Wassim, je vous remercie tous les trois pour votre considération : vos remarques sont une des principales raisons du succès de mon doctorat. Olivier, je vous remercie pour votre support illimité : votre expérience dans le codage d'images a été toujours utile pour faire fonctionner les algorithmes et obtenir de bons résultats, et nos longues discussions m'ont vraiment aidée à prendre les bonnes décisions.

Je remercie de même Pr. Jean-Chrisptophe Burie et Pr. Chaouki Diab pour la révision de cette thèse, et Dr. Vincent Ricordel, Pr. Luce Morin et Dr. Bachar El Hassan pour leur participation au jury.

Je remercie très vivement l'ensemble du personnel de l'Institut d'Electronique et de Télécommunications de Rennes et du centre AZM pour la Recherche en Biotechnologies et ses Applications, qui procurent un environnement de travail propice et chaleureux.

Je tiens à exprimer mes remerciements aux personnes avec qui j'ai eu le plaisir de travailler. Merci aux anciens doctorants m'avoir introduire dans le monde du LAR : François Pasteau et Emilie Bosc. Nos débats ont toujours été un plaisir pour moi. Merci à tous mes collègues avec qui j'ai aimé travailler : Yi Liu, Khaled Jerbi, Mariam Abid, Wassim Hamidouche. Je tiens à remercier chaleureusement Pr. Luce Morin, qui a offert beaucoup de son temps pour l'élaboration de ce travail. Je voudrais également remercier tout spécialement Mme Jocelyne Tremier, Mme Corinne Calo et Mme Jana El Hajj pour la gestion des tâches administratives de façon transparente.

Je tiens également à remercier l'équipe du Centre Numérique de la Francophonie qui, derrière son chef Dr. Hassan Amoud, m'a accueillie surtout pour faire les réunions sur *skype*.

En plus, je tiens à remercier l'association AZM pour le financement de ce travail en collaboration avec l'Université Libanaise.

Enfin mes derniers remerciements, qui n'en sont pas pour autant les moins importants, vont à ma famille et mes amis. Un grand merci à mes parents (Mariam et Jamal) pour leur confiance, ma sœur Nour pour sa patience au quotidien, et mon frère Lieutenant Ahmad pour son soutien illimité. Merci beaucoup à tous mes amis : merci pour les très bons moments que nous avons passés et pour les incroyables soirées que nous avons vécues ensemble.

TABLE DES MATIÈRES

1 INTRODUCTION GÉNÉRALE

Avoir le savoir est si courant,
en prendre soin est si rare.

— Ali Ibn Abi Taleb.

HISTOIRE

Au cours de l'histoire, la vision humaine fut au cœur de l'attention aussi bien de grands philosophes que de nombreux chercheurs. Après avoir cru pendant de longs siècles que l'œil était un émetteur de lumière, Ibn Al-Haytham (connu sous son surnom Alhazen), un des savants arabes les plus influents dans le domaine de l'optique, arrive à concevoir, au 10ème siècle après J.C, que l'œil puisse être un récepteur de la lumière réfléchie sur les surfaces des objets. L'œil, ce formidable instrument optique, constitue donc le lien principal entre le monde extérieur et la perception visuelle. La couleur (la texture), la forme, la distance des objets à l'œil (la profondeur) et la distance relative entre les objets eux même (le relief) sont les résultats de l'interprétation des données fournies par l'œil au cerveau, moteur central de traitement des données.

Cette sensation de profondeur et de relief excitait la curiosité des scientifiques depuis l'antiquité. Comment perçoit-on, à partir d'une vue à deux dimensions, la profondeur des objets ? En effet, la vision binoculaire, ou stéréoscopie, est un des facteurs essentiels qui contribue à la perception du relief. Ainsi, la légère disparité entre les deux vues d'une scène, reçues par chacun de nos yeux, permet au cerveau de concevoir la profondeur des objets dans la scène. La profondeur, cette troisième dimension, continue d'inspirer un grand nombre de chercheurs et d'ingénieurs, notamment dans le domaine de cinéma. La reproduction de la sensation de relief et de profondeur est c'est ce qu'on appelle **l'expérience** 3D.

Grâce aux progrès scientifiques, notamment vers la fin du 20ème siècle, la technologie 3D connaît un véritable succès avec l'invention des écrans stéréoscopiques numériques. En 2009, la première télévision en relief ou télévision à trois dimensions (3DTV) a été commercialisée. Ces écrans simulent la vision binoculaire pour reproduire la sensation de profondeur à l'observateur (voir FIGURE 1).

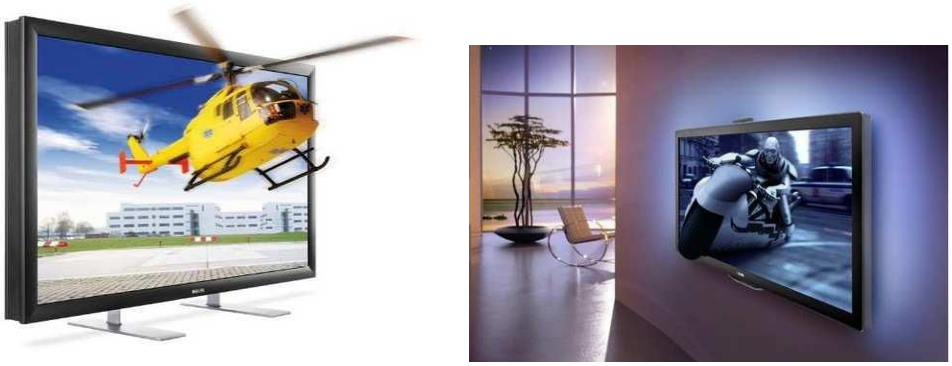


FIGURE 1. Exemple de l'impression de profondeur devant une 3DTV de Philips.

MOTIVATIONS

Récemment, presque toutes les disciplines se sont tournées vers l'utilisation de la technologie 3D, notamment le biomédical, les jeux de divertissement, les documentaires ou encore la cinéma. Cette diversité de disciplines entraine une hétérogénéité dans la qualité et les fonctionnalités

demandées de la technologie 3D. En d'autres termes, certaines applications, telles que le biomédical, nécessitent une haute qualité 3D avec des fonctionnalités plus avancées en comparaison à d'autres applications telles que les consoles des jeux.

En outre, l'énorme quantité de données disponibles exige des traitements automatisés, et par là-même le développement de techniques avancées pour l'indexation d'images, la reconnaissance d'objets, etc. L'efficacité de telles applications est directement liée au niveau sémantique extrait de l'image : plus la représentation de l'image est précise, plus l'interprétation est performante, et inversement.

A titre illustratif, la plateforme "3D end-to-end" constitue une chaîne complète classique d'acquisition, de codage, de transmission, de synthèse et enfin d'affichage pour les utilisateurs (voir FIGURE 2). La phase de codage 3D est critique dans la chaîne 3D, dans la mesure où elle doit être adaptée au format d'acquisition, suffisamment performante pour respecter les contraintes de débit fixées par le canal de transmission, et influe le plus significativement sur la qualité des images 3D obtenues après synthèse de vues.

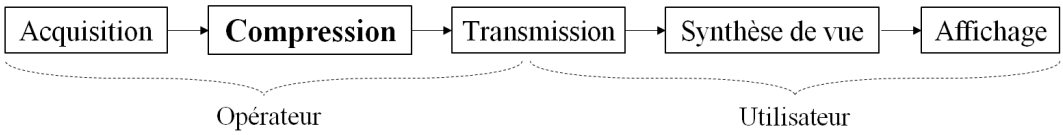


FIGURE 2. Plateforme globale d'un système 3D "end-to-end".

Un des principaux objectifs lors de la conception d'un schéma de codage d'images 3D est ainsi de trouver une solution qui 1) réponde à cette hétérogénéité de qualité, et 2) soit capable si possible d'extraire la sémantique de l'image 3D. Ceci nécessite deux éléments majeurs.

1) Un schéma scalable capable d'adapter le flux de données au canal de transmission et à la capacité du récepteur. Outre la scalabilité, il est nécessaire que le schéma de codage d'images 3D préserve la cohérence entre la texture et la profondeur de la scène. Les codeurs standards d'images 3D considèrent le plus souvent indépendamment le codage des composantes de texture et de profondeur, ce qui entraîne un manque de cohérence pour les images reconstruites. De plus, ces codeurs ont été en général conçus initialement pour la compression de la texture, et réutilisés ensuite pour la profondeur, alors que les deux types de composantes présentent des caractéristiques très différentes.

2) Un codage d'images 3D capable d'extraire la sémantique de l'image 3D en intégrant des outils d'analyse pour une interprétation du contenu. Un modèle intuitif pour aider à cette interprétation est celui qui considère une scène comme une compositions de régions et d'objets. Une telle description doit évidemment être flexible afin de concevoir une description interprétable. Les représentations en régions peuvent être envisagées à différents niveaux de granularité, depuis des représentations détaillées comprenant de nombreuses régions, à des représentations plus simples où une région peut être associée à un objet. De telles approches permettent donc de combler le fossé entre les systèmes informatiques et le Système Visuel Humain (SVH). Malheureusement, les méthodes de représentation en régions existantes permettent d'extraire uniquement les formes des objets de la scène indépendamment du codage de leur contenu. Par exemple, pour toutes les techniques classiques de compression par blocs, il ne peut y avoir correspondance complète entre forme des régions et codage de leur contenu.

Ainsi, différents défis se posent dans du codage d'images 3D. En particulier, il s'agit de demander comment concevoir un schéma de codage scalable qui préserve la consistance entre la profondeur et la texture, et offre dans le même temps une meilleure représentation de la scène ? Une autre question connexe peut être posée, à savoir comment coupler de façon cohérente une représentation fine des contours avec un schéma de codage du contenu des images ?

OBJECTIFS

L'objectif de cette thèse n'est donc pas de proposer ni une "simple" méthode de codage 3D, ni une méthode de représentation de la scène par segmentation, mais plutôt une solution jointe de représentation et de compression dédiées aux images 3D. Il s'agit ainsi de tenter de répondre

aux problèmes propres à la compression d'images 3D à haut niveau sémantique tels que l'hétérogénéité de la qualité après synthèse de vue, la cohérence entre composantes de profondeur et de texture, la fusion d'un schéma global de représentation fine et de codage région dans une scène 3D. En d'autres termes, il s'agit de développer un schéma scalable et joint texture/profondeur avec des fonctionnalités avancées. Les techniques mises en œuvre s'appuient sur le codec LAR (*Locally Adaptive Resolution*), initialement conçu pour le codage d'images 2D. Le LAR se base sur le principe d'adapter le niveau de compression en fonction de l'activité locale de l'image afin de s'adapter au SVH. Plusieurs fonctionnalités sont intégrées au LAR, notamment un algorithme de segmentation 2D à coût nul et un codage par Régions d'Intérêt (RoI).

Plusieurs axes de recherche ont été développés afin d'étendre les principes du codage LAR à la 3D. Ils peuvent se résumer comme suit :

- le codage 3D à bas débit préservant les contours principaux des objets,
- le rehaussement spatial de la texture pour un codage scalable,
- la segmentation sémantique 3D pour une représentation fine des objets,
- "l'Autofocus 3D" pour un schéma joint de représentation fine et de codage basé contenu d'images.

ORGANISATION DU DOCUMENT

Cette thèse se divise en deux parties principales qui sont la compression d'images 3D, et le codage par régions d'intérêt 3D. Chaque partie est subdivisée en chapitres présentés ci-après.

La première partie de cette thèse concerne donc les aspects compression 3D. Le Chapitre 2 introduit plus en détail la plateforme 3D "end-to-end", depuis l'opérateur jusqu'à l'utilisateur.. La compréhension de cette plateforme impose ainsi une liste de contraintes qui doivent être prises en compte pour la compression des données 3D. Dans le Chapitre 3, nous analysons les différents codeurs standards 2D, leurs extensions au contexte 3D, les propriétés des cartes de profondeur ainsi que les différentes méthodes de compression de profondeur de l'État de l'Art qui leur sont adaptées. Le schéma de codage **joint** 3D **scalable** proposé est ensuite introduit et évalué dans le Chapitre 4.

La deuxième partie de cette thèse concerne les aspects codage par régions d'intérêt 3D. Le Chapitre 5 décrit et analyse les différentes approches existantes de segmentation 3D de l'État de l'Art. Enfin, nous introduisons dans le Chapitre 6, deux solutions pour un schéma global de représentation fine des contours et de codage basé contenu d'images. La première solution est appelée "Autofocus 3D simple" ne nécessitant pas de phase de segmentation. La seconde solution dite "Autofocus 3D avancé" s'appuie en supplément sur une segmentation sémantique de la scène 3D. Ces deux solutions sont évaluées sur les images 3D de référence fournies par MPEG.

La FIGURE 3 illustre le plan des deux parties de cette thèse.

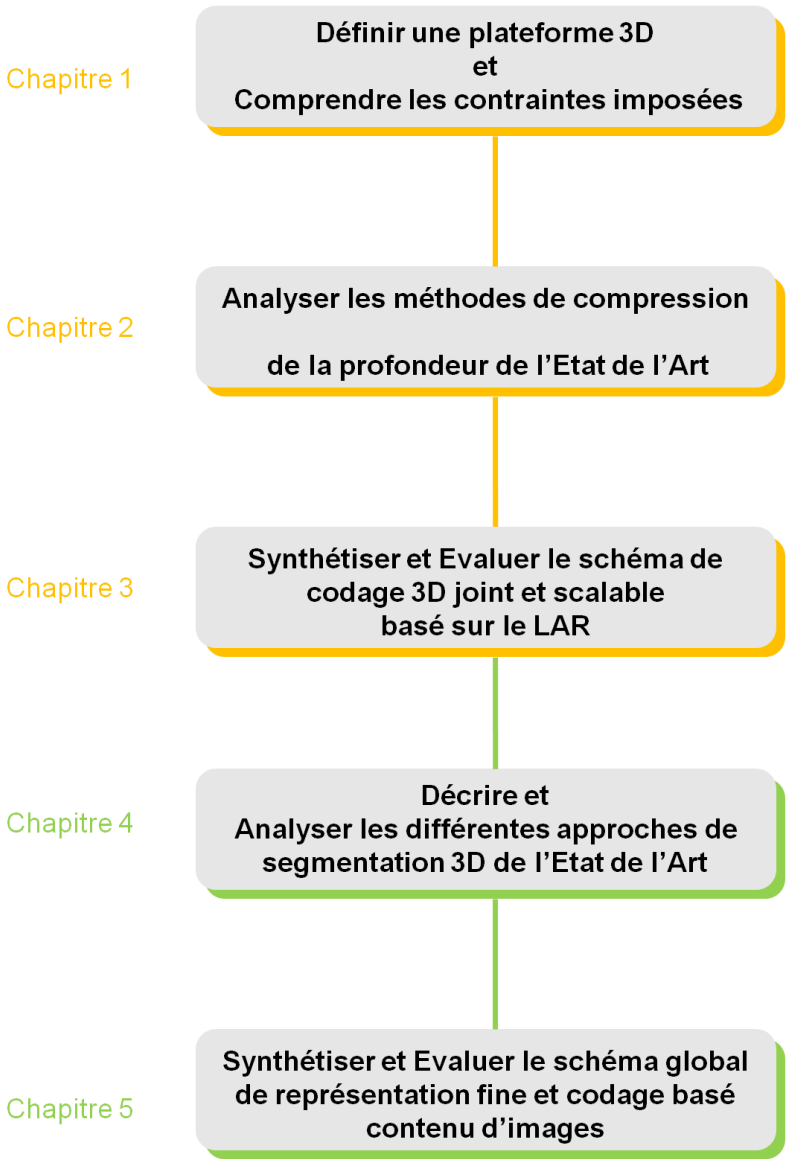


FIGURE 3. Organisation du document.

Première partie

COMPRESSION 3D

ÉTAT DE L'ART

2 PLATEFORME 3D

Cela pourrait jouer un rôle important
dans la résolution de crimes.
La capacité de voir en 3D la scène de crime
donne aux jurés plus de vraisemblance que
ne pourrait le faire les représentations 2D.

— Jordan Crook, journaliste à TechCrunch

Objectifs spécifiques du chapitre :

- **Connaître** la plateforme 3D.
- **Comprendre** les contraintes imposées par cette plateforme.

2.1 INTRODUCTION

"Voir le relief, c'est recevoir, au moyen de chaque œil, l'impression simultanée de deux images dissemblables du même objet", EUCLIDE au $3^{\text{ème}}$ siècle avant J.C. De nos jours, en se basant sur ce principe de disparité binoculaire, les nouvelles technologies offrent aux spectateurs une troisième dimension pour fournir l'impression de profondeur et une visualisation en relief.

Cette technologie s'intègre dans presque tous les domaines tels que la capture des jeux sportifs, les documentaires[1], le divertissement (e.g. cinéma 3D, télévision 3D 3DTV, *Free view point TV*FTV, jeux numériques 3D Nintendo 3DS [2]), le génie biomédical (e.g. modélisation géométrique et biomécanique 3D des structures biologiques[3]), et récemment les enquêtes criminelles (e.g. le département de la police de Roswell, NM en Etats Unis, a récemment utilisé l'appareil *Focus*3D *X* 300 conçu par la société *Faro*[4] pour mieux analyser les scènes de crimes[5]).

Cette diversité d'applications implique une diversité de formats et de codage des données 3D en fonction du type d'acquisition, de la bande passante de la chaîne de transmission et du niveau de qualité et de précision de l'affichage.

Afin de positionner les contraintes des différentes applications basées sur le système 3D, nous décrivons dans ce chapitre les différentes phases du système 3D illustrées dans la FIGURE 4.

La plateforme globale du système 3D consiste dans un premier temps, en une phase d'acquisition pour générer les données 3D sous différents formats. Ensuite, suivant le format adopté, ces données sont codées pour être transmises. Puis, après décodage, la phase de synthèse de vues virtuelles consiste à créer certaines vues intermédiaires ou virtuelles différentes de celles issues du banc d'acquisition. Enfin, un système 3D offre aux utilisateurs un affichage en 3D avec l'impression de profondeur et/ou affichage mutli-vues à l'aide des écrans stéréoscopiques (avec lunettes) ou auto-stéréoscopiques (sans lunettes). De même, il permet de désactiver l'affichage 3D pour conserver l' affichage classique 2D.

La Section 2.2 introduit les différents formats d'acquisition de la plateforme 3D. La Section 2.3 cite ensuite les caractéristiques et les différents concepts de codage 3D. Le principe de synthèse de vue est expliqué dans la Section 2.4. la Section 2.5 présente ensuite les différentes technologies d'affichage 3D. La Section 2.6 présente certaines fonctionnalités avancées dans le domaine de l'image. Enfin, la Section 2.7 résume les contraintes impliquées par chaque phase de la plateforme 3D.

1 http://www.binocle.com/?-Documentaires-
2 http://www.nintendo.com/3ds/features/#/3d-features
3 http://www.etsmtl.ca/recherche/chaires-unites-rech/Chaires/CRC-imagerie-3D
4 http://www.faro.com/en-us/products/3d-surveying/faro-focus3d/overview
5 http://techcrunch.com/2014/01/27/police-using-3d-scanners-for-panoramic-crime-scene-analysis/

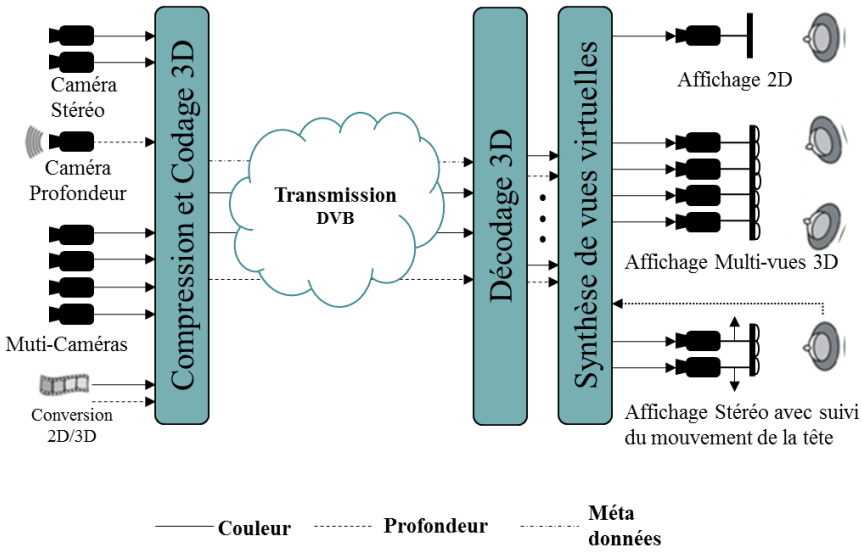


FIGURE 4. Plateforme 3D de l'acquisition à l'affichage.

2.2 ACQUISITION

2.2.1 *Acquisition binoculaire basée image*

L'acquisition binoculaire basée image est effectuée avec l'utilisation de deux caméras ordinaires parfaitement calibrées et synchronisées, et possédant les mêmes paramètres optiques (focale, distance de mise au point, temps d'exposition, etc.). Ces deux caméras sont fixées sur un dispositif appelé *rig*, qui est : -soit rigide avec une distance fixe entre elles (entraxe) et généralement égale à la distance inter-oculaire [6], -soit articulé (robotisé) avec un entraxe réglable, (voir FIGURE 5).

FIGURE 5. Exemple de systèmes d'acquisition binoculaire basés image : (a) rig rigide ; (b) rig robotisé héliporté utilisé par Binocle pour le film "La France entre ciel et mer".

Ce dispositif fournit un couple stéréo (i.e. un couple d'images gauche et droite avec une légère disparité). Un exemple d'un couple stéréo est donné dans la FIGURE 6. Ces images sont sujettes à des post-traitements dans un contexte de postproduction stéréoscopique tels que la rectification [1, 2] ou l'égalisation colorimétrique (puisque les deux images peuvent avoir une légère différence de luminosité lors de la capture) [3, 4, 5].

Les inconvénients de ce format d'acquisition sont la redondance d'information entre les deux images du couple stéréo due à la légère disparité entre les deux caméras, la dépendance de

6 distance séparant les deux yeux de l'être humain, à peu près 6.5 cm

FIGURE 6. Couple d'images vues par la caméra gauche (a) et la caméra droite (b) de la même scène.

l'affichage des conditions d'acquisition (la ligne de base par exemple) et la complexité des algorithmes de post-traitements.

2.2.2 *Acquisition binoculaire basée profondeur*

Au lieu de représenter une scène par deux images couleur, elle peut être représentée par sa géométrie et une seule image couleur. Cette géométrie correspond à la distance à la caméra des objets dans la scène, et est appelée la **profondeur**. Une telle représentation est appelée "2D plus profondeur" ou 2D+Z (*2D-plus-depth*). A l'aide de information 2D+Z, on peut extrapoler une deuxième vue à une position différente dans l'espace (voir Section 2.4). La deuxième vue couleur (extrapolée) et la vue originale constituent alors un couple stéréo. Plusieurs méthodes sont disponibles pour trouver la profondeur des objets dans la scène.

Une première méthode consiste à reconstruire la carte de profondeur à partir du couple capturé d'images stéréo, [6]. Un système simplifié de stéréovision[7] est illustré dans la FIGURE 7. Les variables de la FIGURE 7 sont les suivantes :

- b est la distance entre les deux caméras (ligne de base),
- f est la distance focale des caméras,
- X_k avec $k = \{A, B\}$ est l'axe des X, respectivement de la caméra gauche et droite,
- Z_k avec $k = \{A, B\}$ est l'axe optique, respectivement de la caméra gauche et droite,
- P est un point réel défini par les coordonnées X, Y, et sa *profondeur* Z,
- u_L est la projection du point réel P en une image acquise par la caméra de gauche,
- u_R est la projection du point réel P en une image acquise par la caméra de droite,

Les deux caméras étant séparées d'une distance b, le point réel P est vu par deux perspectives différentes. Les abscisses des points u_L et u_R sont définies par :

$$\begin{aligned} u_L &= f * \frac{X}{Z} \\ u_R &= f * \frac{X - b}{Z} \end{aligned} \tag{1}$$

La différence de position du point réel P entre l'image droite et l'image gauche correspond à la disparité d :

$$d = u_L - u_R = f * \frac{b}{Z} \tag{2}$$

7 http ://www.ni.com/white-paper/14103/fr/

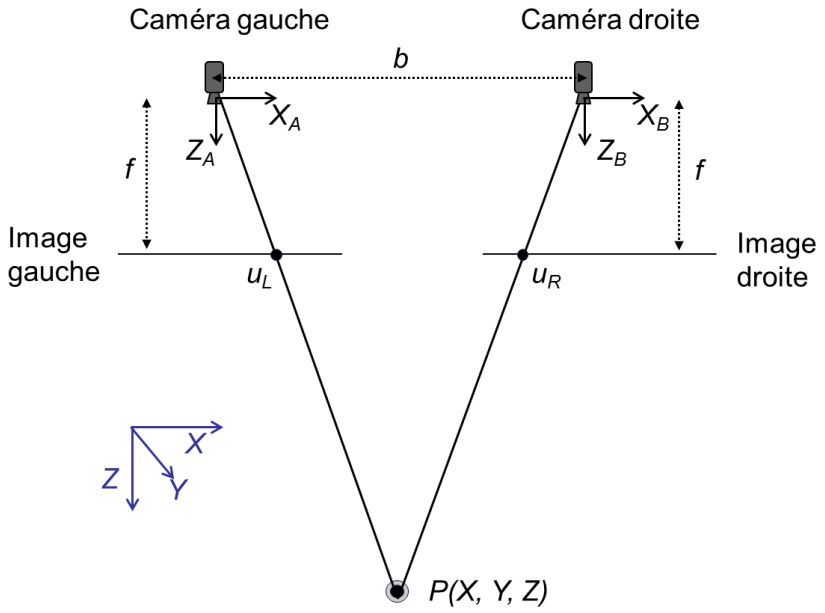


FIGURE 7. Système simplifié de stéréovision

Enfin, la profondeur réelle Z qui correspond à la distance entre le point réel *P* et la ligne de base, est calculée en fonction de la disparité :

$$Z = f * \frac{b}{d} \tag{3}$$

Ainsi, $Z \in [Z_{proche}, Z_{loin}]$, où Z_{proche} et Z_{loin} sont deux valeurs > 0 qui représentent respectivement les plans de délimitation le plus proche et le plus éloigné de la caméra et dépendent évidemment de la scène. Les valeurs de profondeurs réelles Z sont échelonnées uniformément entre 0 et 255 afin d'obtenir une image ou une carte de profondeur D en niveau de gris ($D \in [0, ..., 255]$). Cette carte de profondeur représentent les objets, avec la convention que les plus éloignés sont en noir, et les plus proches sont en blanc. Cette conversion entre la profondeur réelle et la carte en niveau de gris peut être soit linéaire (voir Eq. 4) soit non linéaire (voir Eq. 5), [7].

$$D = \left\lfloor 255. \frac{Z - Z_{loin}}{Z_{proche} - Z_{loin}} \right\rfloor \tag{4}$$

$$D = \left\lfloor 255. \left(\frac{1}{Z} - \frac{1}{Z_{loin}}\right) \Big/ \left(\frac{1}{Z_{proche}} - \frac{1}{Z_{loin}}\right) \right\rfloor \tag{5}$$

Ainsi à chaque pixel(x,y) de l'image couleur correspond un pixel dans la carte de profondeur représenté sur 8 bits où la valeur minimale 0 et la valeur maximale 255 représentent respectivement la profondeur Z la plus proche et la plus éloignée.

L'avantage de la transformation non linéaire par rapport à transformation linéaire, est d'assurer une meilleure résolution pour les profondeurs proches [7].

Plusieurs systèmes d'acquisition commerciaux intègrent cette méthode de stéréovision pour calculer la géométrie de la scène. Par exemple, Videre design[8] propose des têtes stéréo à entraxe fixe ou variable, avec un calcul de la disparité/profondeur par logiciel (*Small Vision System*). La tête stéréo Tyzx DeepSea [9], proposée avec plusieurs options d'entraxe, utilise un FPGA embarqué pour le calcul de la carte de profondeur.

Cette méthode de calcul de la géométrie de la scène présente certaines incertitudes dans le calcul de la profondeur des objets dans la scène. En effet, l'opération d'estimation de la disparité

8 http ://users.recn.com/mclaughl.dnai/.
9 http ://tyzx.com/products/camers.html

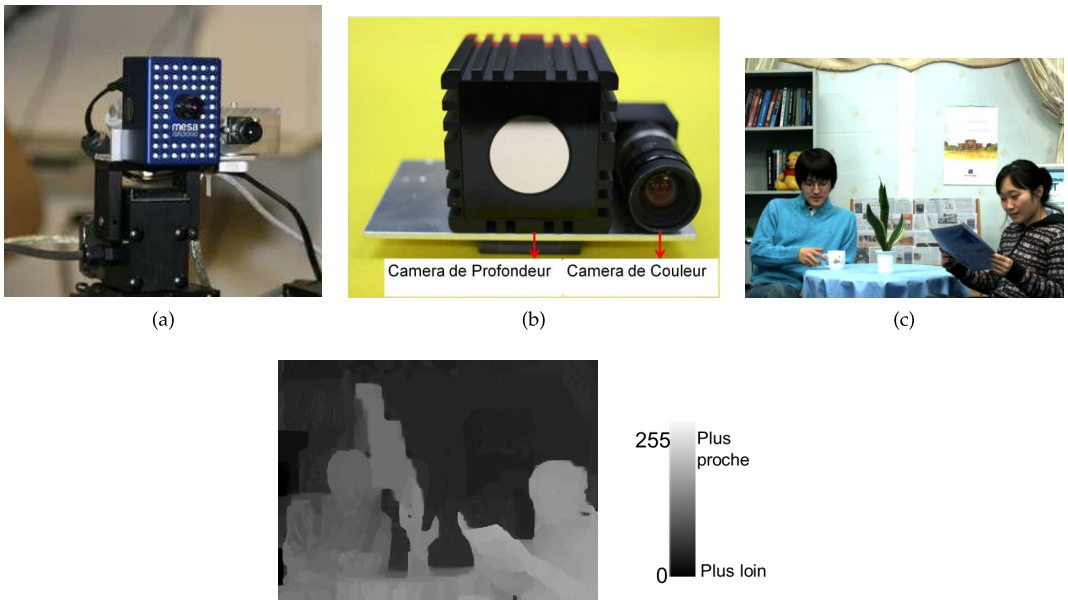
affronte plusieurs problèmes tels que le problème d'ambiguïté dû à la variation de l'illumination et du contraste entre les images gauche et droite, le problème des zones occultées, des discontinuités et des motifs répétitifs [8].

Une autre méthode plus efficace est d'utiliser une caméra de profondeur appelée *Time-of-Flight camera ToF*[10]), illustrée dans la FIGURE 8, pour la reconstruction de la carte de profondeur : des faisceaux infrarouges sont émis de la caméra vers la scène, puis les rayons réfléchis sont collectés par la caméra pour mesurer le temps d'aller-retour. En tenant compte de ce temps et de la vitesse des faisceaux d'infrarouges, la distance entre les objets et la caméra peut être calculée, et la carte de profondeur est ainsi reconstruite. L'avantage d'un tel type de caméra est qu'il évite les problèmes d'illumination, d'occultation et il est adéquat pour les applications en temps réel. Alors qu'il présente certaines limitations : la carte de profondeur générée par la *ToF* est bruitée à cause des interférences probables avec les rayons infrarouges émis [9]. La fameuse *Kinect* par exemple, utilisée dans les jeux vidéos, utilise un capteur de profondeur de type *ToF* pour détecter le mouvement des utilisateurs (voir FIGURE 8.c).

(a) (b) (c)

FIGURE 8. Exemples de caméras de profondeur time-of-flight : (a) SwissRanger SR3000 de Mesa Imaging ; (b) CamCube 2.0 de PMD Technologies, (c) exemple de Kinect XBOX 360 de Microsoft.

La FIGURE 9 donne des exemples de systèmes d'acquisition binoculaire basés profondeur [10, 11].



(a) (b) (c)

(d)

FIGURE 9. (a) et (b) Exemples du système d'acquisition binoculaire basé profondeur fournissant (c) une image couleur et (d) la carte de profondeur associée.

10 http ://www.fotonic.com/content/Products/fotonic-products-c-series.aspx

2.2.3 *Acquisition multivues basée image*

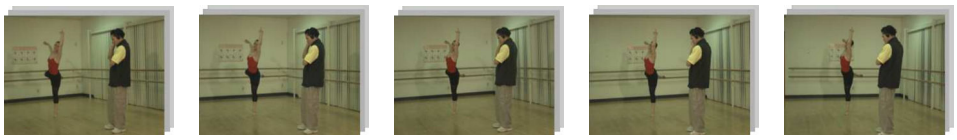
Pour augmenter l'expérience 3D et apporter une impression d'immersion 3D, la scène peut être capturée selon plusieurs points de vue. Les systèmes d'acquisition multivues basée image consistent en une capture de données vidéos synchronisées représentant différents points de vue d'une même scène (*MultiView Video* MVV). Ces systèmes utilisent plusieurs caméras avec divers arrangements et répartitions selon l'application visée.

Des systèmes multivues latéraux ou directionnels utilisent des caméras (intégrées ou assemblées) réparties de façon régulière sur une courbe (rectiligne ou non) (voir FIGURE 10a) ou une grille (plane ou non) (voir FIGURE 10b). Chaque paire de caméras forme un couple stéréo local. De tels dispositifs fournissent des points de vue proches les uns des autres, ce qui est utile pour les applications telles que la visualisation en relief et/ou la navigation libre (FTV). Un exemple de données MVV est donné dans la FIGURE 11.

(a) Cam-box : prototype de caméra 8 points de vue intégrée développée par 3DTV Solutions et l'Université de Reims, France

(b) Grille de caméra 100 points de vues développé par l'Université de Nagoya, Japon

FIGURE 10. Exemples de systèmes d'acquisition multivues latéraux [12].



(a) Vue 4 (b) Vue 3 (c) Vue 2 (d) Vue 1 (e) Vue 0

FIGURE 11. Exemple de données MVV : Image 14 de la Séquence Ballet Dancer (Images fournies par Microsoft Research).

Des systèmes multivues englobants ou omnidirectionnels utilisent des caméras relativement espacées et approximativement convergentes de sorte à couvrir la scène, (voir FIGURE 12, [12]). De tels systèmes sont principalement destinés au "bullet time[11]" (caméra virtuelle se déplaçant à temps ralenti) à large secteur angulaire.

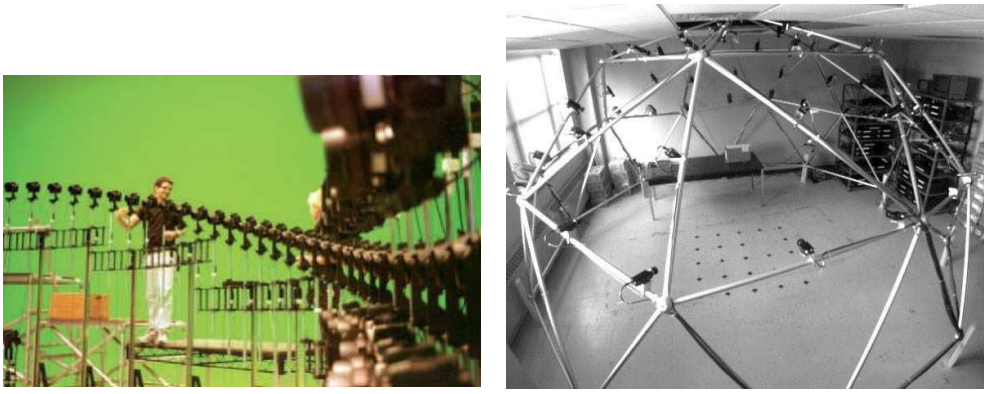
Les post-traitements (calibration, correction colorimétrique, rectification...) sont également appliqués aux images capturées par le système d'acquisition multivues basé image.

2.2.4 *Acquisition multivues basée profondeur*

L'acquisition multivues basée profondeur peut être considérée comme une combinaison de l'acquisition 2D+Z (acquisition binoculaire basée profondeur) avec l'acquisition MVV (acquisition multivues basée image). C'est ainsi un ensemble de vidéos 2D capturées avec leur carte de profondeur associée (*Multiview Video plus Depth* MVD), voir FIGURE 13.

Un tel système acquiert une grande importance puisqu'il réduit la densité des caméras 2D sur l'axe ou dans la grille d'acquisition (chaque paire de caméras stéréo est remplacée par une

11 https ://www.youtube.com/watch ?v=QM4tbYNv6KU



(a) Système d'acquisition avec 120 caméras pour le tournage de Matrix Warner Bros (effet bullet time)

(b) Système d'acquisition englobant avec 51 caméras dévelopé par l'Université Carneige Mellon

FIGURE 12. Exemples de systèmes d'acquisition multivues englobants [13].

seule caméra 2D et une caméra profondeur), et donc permet d'élargir l'angle de vue du système d'acquisition.

(a) Vue 4 (b) Vue 3 (c) Vue 2 (d) Vue 1 (e) Vue 0

FIGURE 13. Exemple de données MVD : à chaque point de vue, une image texture (couleur) associée à une carte de profondeur (Image 14 de la Séquence Ballet Dancer (ensemble d'images fourni par Microsoft Research)).

2.3 COMPRESSION

L'acquisition des vidéos binoculaires (deux vues) ou multivues (N vues) génère un volume de données de plus en plus lourd en termes de mémoire, notamment avec l'apparition actuelle de la norme *Ultra High Definition* (très haute résolution) ou format 4K (3840 pixels $\times$ 2160 lignes). Ceci induit la nécessité d'une phase de compression des données 3D au sein de la plateforme 3D. Dans cette section, nous décrivons les principales caractéristiques d'un codec 3D. Dans le prochain chapitre, nous classifions et analysons d'un point de vue global, les différentes méthodes utilisées pour la compression 3D suivant le type de données 3D.

2.3.1 *Caractéristiques d'un codec 3D*

Les codeurs/décodeurs 3D possèdent quatre caractéristiques principales : 1) le caractère standard ou non-standard, 2) la compatibilité avec les codecs 2D, 3) l'optimisation débit-distorsion, et 4) la scalabilité en vues dans le cas d'un système d'acquisition multivues.

1) Standard ou non-standard. Comme dans tous les domaines, il existe des méthodes de codage standardisées. Les deux comités internationaux en charge de la normalisation de nouvelles méthodes de codage sont le "Moving Picture Expert Group" (ISO/IEC MPEG) et le "Video Coding Expert Group" (IUT-T VCEG). Ces deux comités sont parfois rassemblés en comité joint ou "Joint Video Team" (JVT), pour la définition de standards communs comme H.264/MPEG-4

Advanced Video Coding (AVC) [13] ou High Efficient Video Coding (HEVC) [14, 15]. Récemment en mars 2012, un comité joint nommé JVT-3V a lancé un Call for Proposol (CfP) pour des extensions des standards existants aux données 3D [16]. L'intérêt de disposer d'une solution standardisée est de garantir une interopérabilité des services. En revanche, ils ont été défini suivant des spécifications précises, qui ne répondent pas forcément à tous les besoins. Sur les aspects codage par exemple, les standards actuels offrent les meilleures performances de compression d'un point de vue débit/distorsion, mais n'offrent pas forcément un support adapté pour des fonctionnalités telles que des représentations basées contenu par exemple. Des solutions non standard de codage peuvent ainsi être préférées pour des objectifs particuliers de codage.

2) La compatibilité "Backward" et "Forward" (*Backward Compatibility BC* et *Forward Compatibility FC*). Les informations 3D additionnelles (images de plusieurs point de vues, cartes de profondeur, paramètres des caméras) nécessitent des ajouts et des modifications par rapport aux flux de données 2D transmis aux décodeurs. Ainsi, deux types de compatibilité existent : la compatibilité "Backward" et la compatibilité "Forward" (voir FIGURE 14). D'une part, la compatibilité "Forward" est assurée quand l'extension 3D du codeur 2D génère un flux de données 3D qui reste interprétable par le décodeur 2D existant. Ce décodeur 2D peut donc afficher les données 3D transmises. Dans ce cas, les encodeurs, les canaux de transmission, les récepteurs et les décodeurs existants peuvent être réutilisés. D'autre part, la compatibilité "Backward" est assurée lorsque le décodeur 3D est capable de décoder un flux de données de type 2D.

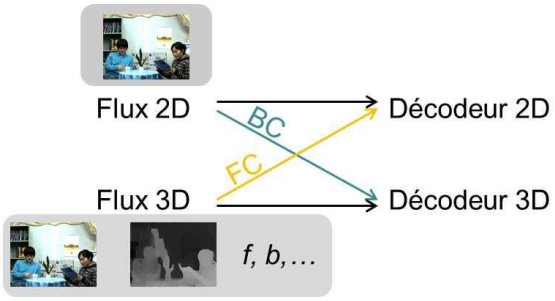


FIGURE 14. Compatibilité "Backward" et "Forward".

3) L'optimisation débit-distorsion. La compression entraîne des distorsions pour les images codées. Ainsi, la problématique classique de la compression d'images est de trouver un compromis entre la qualité des images reconstruites et le débit généré. Dans le contexte du codage des images 3D, il est nécessaire d'optimiser le codage des images 3D, mais aussi potentiellement des images virtuelles synthétisées. Le processus d'optimisation débit-distorsion des codecs 3D est donc plus délicat que celui d'un codec 2D classique.

4) La scalabilité en vues. Dans les prochaines années, les vidéos 3D seront disponibles à travers des applications telles que la télévision broadcast, le stockage de la vidéo, l'Internet en streaming vidéo, et la vidéo sur portable. Cette diversité d'applications implique une nécessité d'interopérabilité des données 3D. Ainsi, la scalabilité en vues est une fonctionnalité qui permet aux flux de données 3D d'être transmis sur des réseaux à capacités variables et d'être affichés sur une multitude de terminaux exigeant différents nombres de vues. En d'autres termes, la scalabilité en vues permet à chaque décodeur de décider le nombre de vues nécessaire à décoder [17]. De plus, différents terminaux exigent non seulement un nombre différent de vues mais encore différentes distances entre les vues. Il s'agit d'une flexibilité de sélection de vues [18].

En outre des différentes caractéristiques d'un codec 3D, plusieurs techniques de codage des données 3D existent. La diversité des formats de données 3D est à l'origine de la multitude de leurs techniques de codage. Dans les sous-sections suivantes, nous discutons justement les différentes méthodes de compression 3D adoptées selon le format de données 3D.

2.3.2 *Méthodes de compression 3D Simulcast*

La première technique adoptée possible pour le codage des données 3D est la technique de codage *simulcast*. Elle consiste simplement à coder les flux vidéos et/ou profondeur de manière

indépendante, en utilisant des codeurs 2D existants, comme le montre la FIGURE 15. Cette technique assure la compatibilité "Backward" et "Forward" ainsi que la scalabilité en vue.

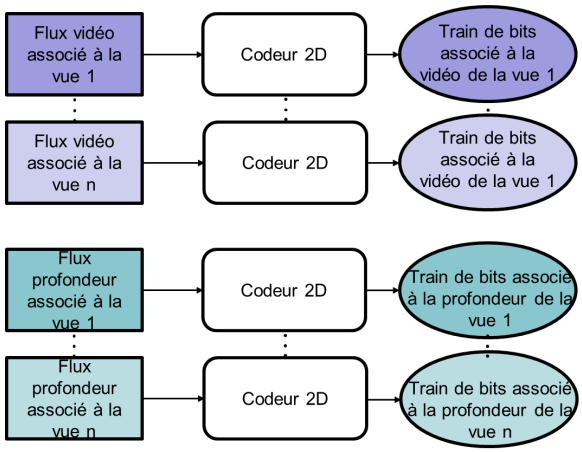


FIGURE 15. Concept de codage *simulcast* des données 3D vidéos et/ou profondeur.

Cependant, cette solution n'exploite pas la redondance existante entre les vues, et conduit à un surcoût de codage. Récemment d'autres concepts ont été adoptés suivant les types d'acquisition des données 3D. L'encodage *simulcast* sert toutefois de référence pour évaluer les autres méthodes.

2.3.3 *Méthodes de compression des données Stéréo et 2D+Z*

Un premier concept concernant le codage des données de type stéréo et 2D+Z est l'utilisation des couches de base et de rehaussement des codeurs 2D. En effet, certains codeurs 2D offrent une fonctionnalité de scalabilité en qualité : une couche de base encode l'image 2D à faible qualité, et une couche, dite de rehaussement, encode un flux auxiliaire qui permet au décodeur de rehausser la qualité de l'image reconstruite à partir de la couche de base. Ainsi pour les données de type stéréo et 2D+Z, la vue de base va constituer la couche de base et l'autre vue, ou la carte de profondeur, constitue la couche de rehaussement pour les données stéréo et 2D+Z, respectivement (voir FIGURE 16). Ensuite, les deux couches sont codées indépendamment et puis les flux générés sont rassemblés par entrelacement temporel. Au niveau du récepteur, le décodeur 2D peut soit se limiter à décoder la couche de base soit à décoder encore la couche de rehaussement. Par suite, un tel concept de codage assure la scalabilité en vue et la rétro-compatibilité avec les décodeurs 2D supportant la scalabilité en qualité.

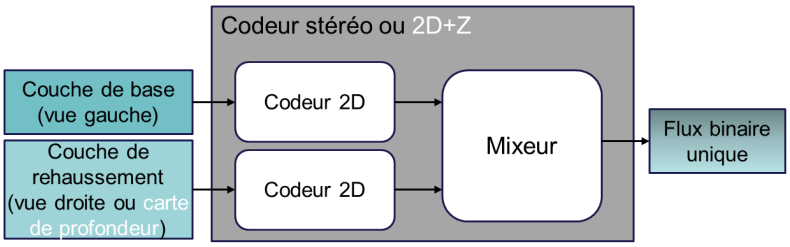


FIGURE 16. Concept de codage stéréo ou 2D+Z utilisant une couche de base et une couche de rehaussement.

Un autre concept de codage des données de type stéréo, appelé *frame compatible* (fc) [19], consiste à garder le nombre d'échantillons des données stéréo égal à celui d'une séquence monoscopique, afin de rester compatible avec la taille des trames 2D. Ainsi, il s'agit de sous-échantillonner et de multiplexer les images des vues gauche et droite en une seule image (multiplexage spatial, voir FIGURE 17) ou en une seule séquence d'images (multiplexage temporel, voir FIGURE 18). En conséquence, les images résultantes peuvent donc être efficacement encodées avec une méthode de compression 2D. L'utilisation du concept *FC* est simple et ne nécessite

pas des modifications sur les infrastructures de distribution. Néanmoins, ces formats réduisent la résolution spatiale ou temporelle ce qui entraîne une perte de qualité. De plus, les décodeurs existants ne sont pas informés du type de multiplexage, et ne peuvent donc plus désentrelacer correctement les données multiplexées. Il en découle une absence de scalabilité en vue et de rétro-compatibilité avec les décodeurs 2D existants [20].

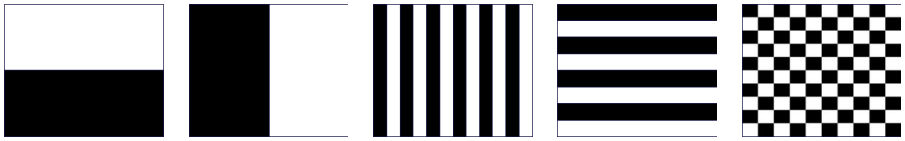


FIGURE 17. Concept de codage stéréo à multiplexage spatial : les images des vues gauches et droite sont sous-échantillonnées, puis combinées dans une image unique. L'entrelacement (a) haut-bas ; (b) côte à côte ; (c) par colonne ; (d) par ligne ; (e) en damier.[21]

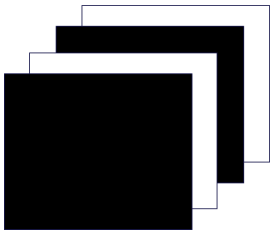


FIGURE 18. Concept de codage stéréo à multiplexage temporel : Les images des vues gauche et droite sont alternativement combinées en une seule séquence.[21]

2.3.4 *Méthodes de compression des données MVV*

Les données multivues étant capturées par des caméras à des positions très proches, une grande redondance existe entre les différentes vues. Les approches de codage de telles données visent ainsi à profiter de la forte corrélation entre les différentes vues pour atteindre des gains de compression élevés. Afin d'assurer un affichage 2D classique sur les terminaux 2D, une vue de base doit être codée indépendamment comme le montre la FIGURE 19.

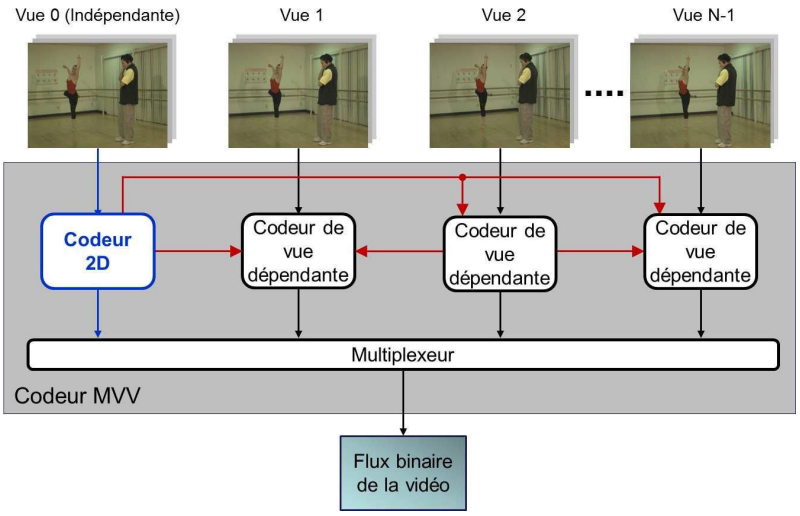


FIGURE 19. Concept de codage MVV exploitant la corrélation inter-vue pour le codage des séquences multivues. Dans cet exemple, la vue 0 est la vue de base.

Au niveau du décodeur, la vue de base peut être décodée seule, alors que toute autre vue ne peut être décodée qu'après la vue de base. Néanmoins, de telles approches n'assurent donc pas une compatibilité "Forward" avec les décodeurs 2D existants. D'autre part, une corrélation négative existe entre la scalabilité en vue et l'efficacité de compression : lorsque la dépendance de codage des différentes vues est fortement exploitée, l'efficacité de codage augmente et la taille du débit généré diminue. En revanche, le décodage d'une vue nécessite le décodage de plusieurs autres vues, ce qui rend la scalabilité en vue moins fine. Inversement, lorsque les différentes vues sont codées avec moins de dépendance, le degré de scalabilité en vue sera plus important. En revanche, la corrélation entre les vues n'étant pas bien exploitée, l'efficacité de codage en terme de débit va diminuer.

2.3.5 *Méthodes de compression des données MVD*

Deux approches sont utilisées pour le codage des données multivues plus profondeur. La première approche consiste à coder les séquences de profondeur en exploitant la corrélation inter-vues entre les cartes de profondeur, mais indépendamment de la texture, comme le montre la FIGURE 20. La deuxième approche profite de la corrélation entre la texture et la profondeur comme le montre la FIGURE 21.

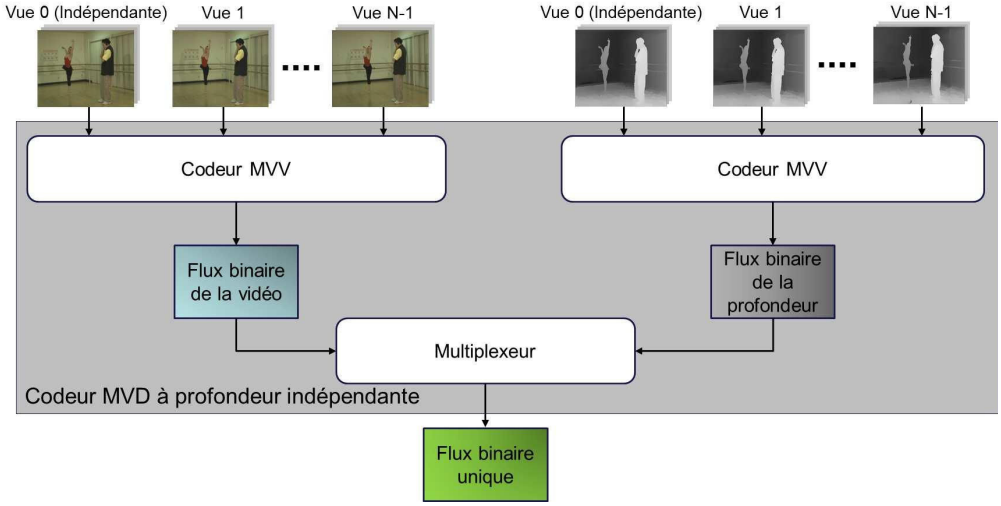


FIGURE 20. Concept du codage MVD codant la profondeur indépendamment de la texture.

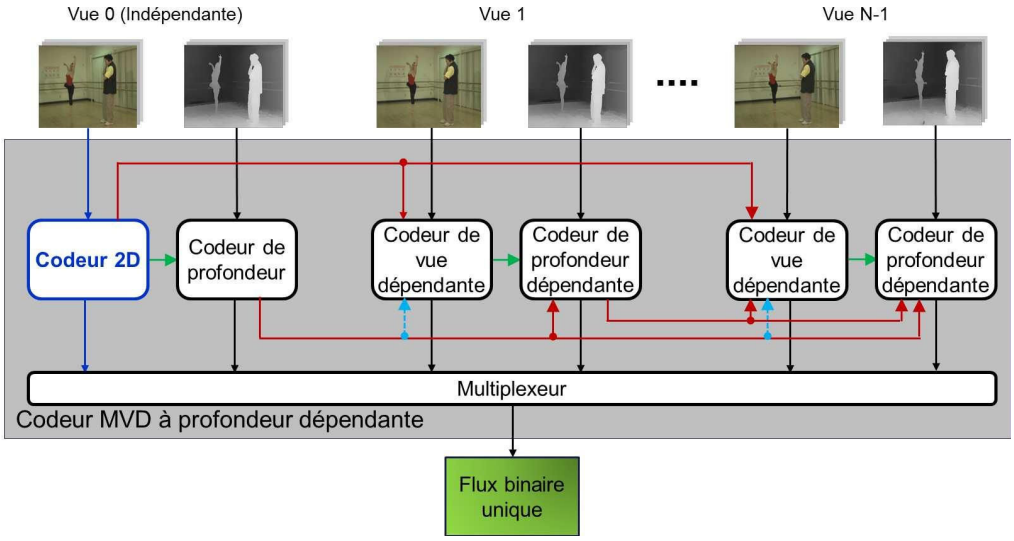


FIGURE 21. Concept du codeur MVD exploitant la corrélation entre la texture et la profondeur.

D'une part, les deux approches n'assurent pas la rétro-compatibilité avec les décodeurs 2D existants. Toutefois, la forte exploitation de la corrélation entre la séquence de texture (ou profondeur) elle même et entre les séquences de la texture et de la profondeur, diminue la granularité de la scalabilité en vue. Les cartes de profondeur ayant des caractéristiques différentes de celle des images naturelles, une étude plus détaillée des différents outils de codage de la profondeur sera donnée dans le chapitre suivant.

2.4 SYNTHÈSE DE VUE

Dans le cas du *Free Viewpoint Video* FVV, certaines vues qui n'existent pas doivent être affichées en temps réel. Il est alors nécessaire de créer par estimation ces vues intermédiaires au niveau du récepteur. Ce processus est appelée la synthèse de vue virtuelle [6] (voir FIGURE 22).

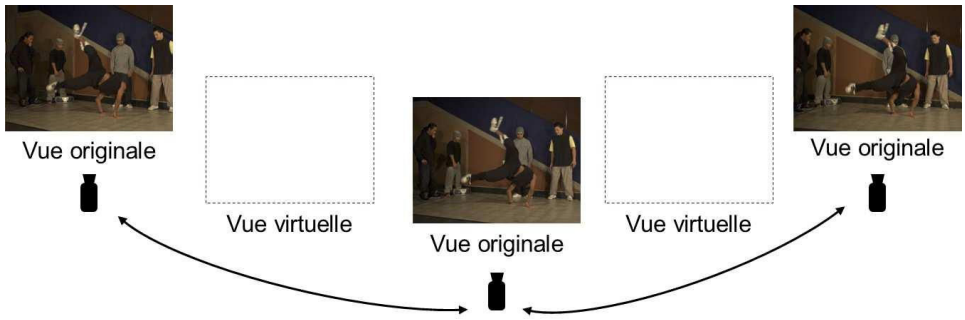


FIGURE 22. Vues virtuelles ou intermédiaires non acquises durant la phase d'acquisition.

Certains algorithmes de synthèse de vue sont basés uniquement sur l'image de texture (*Image-Based Rendering IBR*) [21, 22]. D'autres algorithmes, donnant de meilleurs résultats de synthèse, intègrent l'information de profondeur (*Depth image-Based Rendering DIBR*) [23].

Ces algorithmes sont basés sur les principes de projection d'une image sur un point de vue différent. La projection consiste à estimer les coordonnées d'un point dans l'image originale (i.e. dans le repère de la caméra originale), puis à le projeter dans la vue virtuelle (i.e. dans le repère de la caméra virtuelle) en utilisant les équations mathématiques de transfert de repères [23] (voir FIGURE 23).

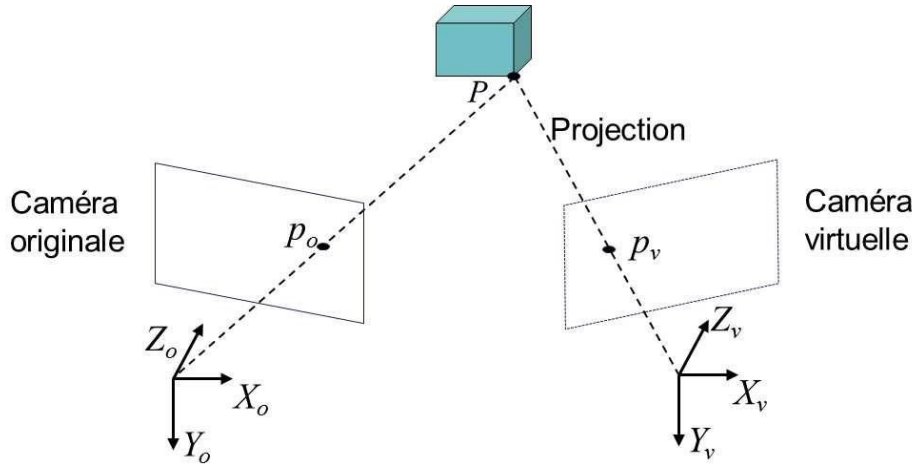


FIGURE 23. Synthèse de vue virtuelle par projection d'un point *P* du plan de la caméra originale sur le plan de la caméra virtuelle.

Cependant, la projection génère plusieurs types d'artéfacts comme illustré dans la FIGURE 24 :

- les contours fantômes provenant de la projection d'un pixel contenant une couleur du fond et une couleur de l'avant-plan (voir FIGURE 24.a) ;
- les craquelures (petites zones découvertes) dues au ré-échantillonnage (voir FIGURE 24.b) ;

- les zones découvertes correspondant aux zones non visibles dans le point de vue original (e.g. objet occulté) qui deviennent visibles dans le point de vue virtuel, en raison de l'effet de disparité (voir FIGURE 24.c) ;

- les artéfacts provenant des erreurs (imprécisions) présentes dans les cartes de profondeur, issues de l'étape de compression.

(a) effet fantôme (b) craquelures (c) zones découvertes

FIGURE 24. Artéfacts associés à la projection lors de la synthèse de vue virtuelle.[28]

Plusieurs approches ont été proposées pour tenter de résoudre ces problèmes. Pour éviter les effets fantômes, *Muller et al* proposent dans [24] de différencier les pixels frontières de l'avant-plan de ceux du fond, et de ne projeter les pixels frontières du fond que s'ils permettent de remplir les zones manquantes. Dans [25], les auteurs proposent un filtrage bilatéral couleur/profondeur pour éliminer les craquelures. Pour le problème des zones découvertes, deux solutions sont possibles. La première implique que la synthèse de vue virtuelle se fasse à partir de plusieurs vues originales proches, et puis fusionner les vues virtuelles obtenues [26]. Une autre solution consiste à remplir les zones découvertes par la technique d'*inpainting* ou "peindre à l'intérieur" qui consiste à diffuser des informations de l'extérieur vers l'intérieur de la zone manquante [27].

Une fois la vue est synthétisée, l'évaluation de sa qualité reste un problème ouvert. En effet, les outils de mesure de qualité objective (PSNR, SSIM) ne sont pas adaptés au système visuel humain et généralement, l'évaluation subjective (ou visuelle) est adoptée. D'autre part, la conception d'outils d'évaluation qui simulent le système visuel humain reste une tâche difficile à développer [28].

2.5 AFFICHAGE

La dernière phase du système 3D est l'affichage des images aux spectateurs sur un écran 3D. Afin d'offrir la sensation de profondeur et d'apporter de l'immersion de l'observateur dans la scène, les écrans 3D se basent sur le principe de disparité pour la création du relief : chaque œil du spectateur doit recevoir une image légèrement différente de l'autre (légère disparité) pour percevoir la profondeur.

2.5.1 *Affichage Stéréoscopique*

Les systèmes d'affichage stéréoscopiques consistent à diffuser les deux images d'un couple stéréo dans un seul faisceau optique indépendamment de la position du spectateur par rapport à l'écran [29]. Par suite, les deux images sont séparées par des lunettes du spectateur (voir FIGURE 25) soit :

- physiquement : filtrage par couleur (rouge et cyan) d'un anaglyphe[12] (voir FIGURE 26a), ou en utilisant des projecteurs et une lunette polarisée (voir FIGURE 26b) ;
- soit temporellement : alternance des vues gauche et droite par l'utilisation d'un écran à haute fréquence d'affichage (*high frame rate display*) et des lunettes à occultations alternées

12 Anaglyphe : image contenant deux images gauche et droite colorées afin affecter une image à chaque œil.

(*synchronized shutter glasses*). Ceci est du au fait qu'à un instant donné, un œil ne voit rien et l'autre reçoit l'image qui lui correspond. Quelques microsecondes après, la situation est inversée (voir FIGURE 26c). Toutefois, une telle technique cause une fatigue des yeux importante, la situation est similaire à une stroboscopie à haute fréquence.

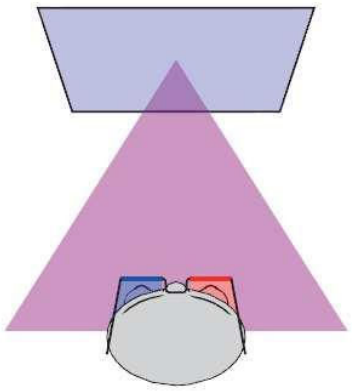
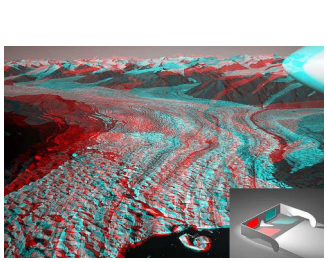


FIGURE 25. Système d'affichage stéréoscopique : un seul faisceau optique transportant deux images puis séparation physique par des lunettes.[21]



(a) séparation par couleur

(b) séparation par polarisation

(c) sépararation temporelle : une 3DTV (200 Hz refresh rate) et lunettes synchronisées par LG

FIGURE 26. Exemples d'affichage stéréoscopique.

2.5.2 *Affichage Auto-stéréoscopique*

Un autre type d'affichage, dit auto-stéréoscopique ou sans lunettes (*glasses free*) [30], permet la séparation des images au niveau de l'écran. Elles sont diffusées dans des faisceaux distincts (FIGURE 27) par un dispositif de lentilles ou de barrière de parallaxe devant l'écran (FIGURE 28). Le spectateur, bien positionné, reçoit simultanément deux images différentes (correspondantes aux deux vues adjacentes du couple stéréo) sur chacun de ses yeux. Les images reçues forment un couple stéréoscopique et le cerveau reconstruit alors le relief par stéréopsie.

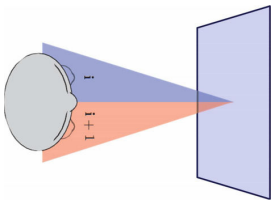
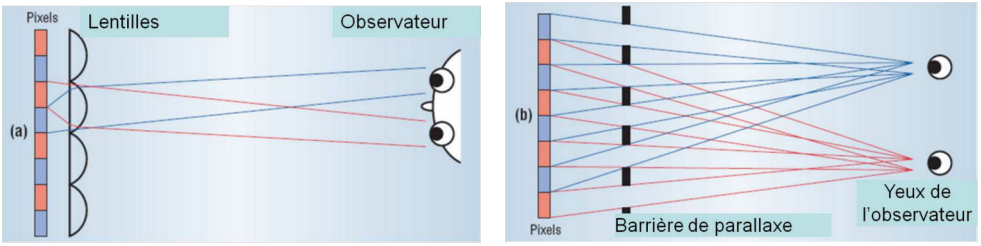


FIGURE 27. Système d'affichage auto-stéréoscopique : deux faisceaux optiques transportant chacun une image [32].



(a) un réseau de lentilles est appliqué devant un écran LCD

(b) une barrière de parallaxe est appliquée devant un écran LCD

FIGURE 28. Construction des écrans auto-stéréoscopiques en plaçant un dispositif devant l'écran LCD afin de (a) dévier ou (b) stopper certains rayons lumineux émis par l'écran en direction de l'observateur [32].

Ce type d'affichage impose que les spectateurs soient placés à une distance et à une position bien déterminées de l'écran 3D, afin de recevoir le bon couple stéréo (voir FIGURE 29). Cette distance optimale est fonction de plusieurs paramètres : la taille de l'écran LCD, sa résolution, et la focale entre l'écran LCD et le dispositif de séparation [31].

Plusieurs travaux récents proposent de nouveaux dispositifs de séparation tolérant le positionnent du spectateur à une distance différente de la distance optimale imposée, en utilisant des barrières dynamiques [32].

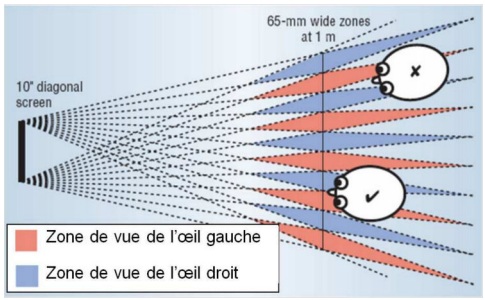


FIGURE 29. Devant un écran auto-stéréoscopique, l'observateur est contraint de se placer à une certaine distance de l'écran 3D et à une certaine position devant l'écran pour une visualisation correcte en relief [32].

2.5.3 *Affichage Auto-multiscopique*

Les écrans auto-multiscopiques diffusent simultanément n vues ($n \geqslant 2$) (l'écran auto-stéréoscopique est un cas particulier de l'écran auto-multiscopique où $n = 2$). Ainsi, des spectateurs placés devant de tels écrans à des positions différentes reçoivent des couples stéréos différentes et par suite des perspectives différentes (voir FIGURE 30). Ces systèmes de visualisation multiscopique permettent à l'utilisateur de se déplacer 180° autour de l'écran et d'avoir la sensation de tourner autour d'un objet visualisé en relief sur l'écran (i.e. l'immersion 3D).

La limitation majeure des systèmes de visualisation récents est que la disparité (*parallax*) est horizontale. Ainsi, le spectateur ne peut pas incliner sa tête ou " s'allonger sur un canapé " [33]. Des prototypes de disparité horizontale et verticale (*full parallax*) ont été récemment développés, présentant des résultats prometteurs [34].

Si le but du système 3D est d'offrir une navigation libre (*Free Viewpoint Video FVV*), un écran 3D consiste simplement en un écran 2D couplé à un système de suivi du mouvement de la tête *Head-tracking*. Ce dispositif permet à l'écran d'afficher la vue qui convient à la position du spectateur par rapport à l'écran[13]. Des exemples d'applications simples de suivi du mouvement

13 https ://www.youtube.com/watch ?feature=player_embedded&v=Jd3-eiid-Uw.

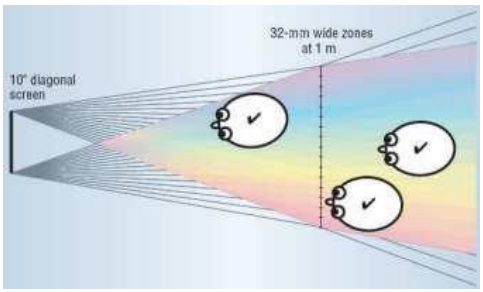


FIGURE 30. Exemple d'affichage auto-multiscopique de 16 vues [32].

de la tête et de navigation libre sont déjà implémentés et téléchargeables sur les *smart phones* (e.g. l'application *i3D*, disponible sur l'*Apple Store*, développée à l'Université de Fourrier[14,15]).

Les affichages stéréoscopiques, auto-stéréoscopiques, et auto-multiscopiques fournissent une sensation de profondeur, alors que d'autres types d'affichage tels que l'affichage volumétrique [35] et holographique [36] fournissent une description détaillée de la scène par le remplissage d'un volume de l'espace 3D par des images ou par reproduction des faisceaux lumineux réfléchis par la scène (voir FIGURE 31). De telles technologies ne sont pas encore matures, il reste beaucoup de problèmes techniques à résoudre.

(a) affichage volumétrique des modèles géométriques - l'Université de Toronto (*Dynamic Graphic Projects*) - et de l'intérieur d'une voiture - *Zebra Imaging*.

(b) hologramme de la chanteuse égyptienne Oum Kolthoum - *ND productions* et *Voxel Animation* en 2012.

FIGURE 31. Exemple d'affichage volumétrique et holographique.

2.6 FONCTIONNALITÉS AVANCÉES

Devant le progrès technologique durant ces dernières années et avec cette évolution explosive de données numériques visuelles d'images et de vidéos, plusieurs questions se posent. Comment faire des recherches dans ces collections d'images le plus efficacement et le plus simplement possible ? Comment exploiter cette immense collection de données ? Pour répondre à ces questions, une solution d'indexation est offerte. Quel est le principe d'indexation ? Quelles sont les

14 http ://blog.laptopmag.com/glasses-free-3d-display-demoed-on-ipad-2.
15 https ://www.youtube.com/watch ?v=bBQQEcfkHoE& feature=player_embedded.

différentes méthodes utilisées pour décrire une image ? On répond brièvement à ces questions, dans les sous-sections suivantes.

2.6.1 *Principe d'Indexation*

L'indexation est un outil récent qui permet de trouver rapidement et efficacement une image dans une base de données. Elle constitue le cœur des systèmes de recherche d'informations visuelles. Un tel système se divise généralement en deux étapes (voir FIGURE 32).

La première étape consiste à indexer ou en d'autres termes à attribuer des descripteurs ou index à chacune des images d'une collection déterminée. Cette première étape est l'étape *offline* car c'est une étape préalable à la recherche.

La seconde étape est la recherche elle-même. L'utilisateur exprime sa requête sous une forme ou une autre : texte, image ou tous les deux, selon la méthode imposée par le moteur de recherche. Ce dernier calcule alors un ou plusieurs descripteurs comme ceux utilisées dans l'étape *offline*. Il compare finalement, grâce à une distance, les descripteurs calculés avec ceux de chaque image de la collection. Le moteur de recherche retourne enfin les résultats sous formes d'images classées par pertinence : du plus ressemblant à la requête, à celui le moins ressemblant. Cette seconde étape est l'étape *online* et est réalisée à chaque nouvelle requête d'un utilisateur du moteur de recherche.

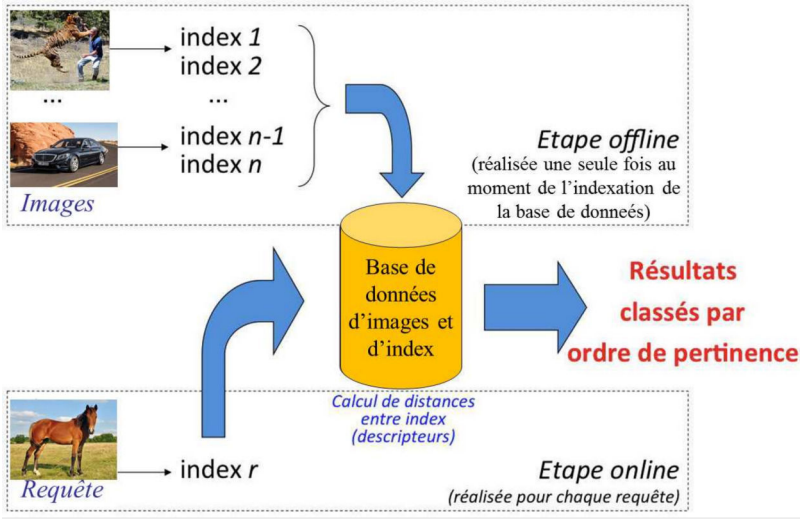


FIGURE 32. Les deux étapes de l'indexation d'images [21].

Dans l'étape *offline* la principale difficulté consiste à trouver les index efficaces pour décrire l'image. Pour l'étape *online*, l'utilisation d'une bonne distance mathématique pour mesurer la similarité entre la requête et les images de la base de données, est le point crucial. Dans la sous-section suivante, nous décrivons les différentes méthodes utilisées pour indexer ou attribuer des descripteurs aux images.

2.6.2 *Méthodes d'Indexation*

L'indexation des images peut être soit manuelle soit automatique. Dans le premier cas, une personne attribue manuellement un ou plusieurs index à chaque image de la base de données. Une telle indexation va être évidemment subjective et va prendre beaucoup de temps. Ainsi, afin d'atténuer le problème de la subjectivité, l'indexation automatique constitue le cœur de la recherche d'informations visuelles. Plusieurs méthodes sont utilisées pour l'indexation automatique des images. Elles peuvent être regroupés sous 3 catégories : l'indexation basée texte, l'indexation basée contenu d'image, et l'indexation basée texte et contenu d'image.

Indexation basée texte : les index attribués à chaque image de la base de données sont des mots-clés tirés du titre de la page contenant l'image (sur le Web par exemple), de la légende

de l'image ou du texte environnant l'image, sans considérer le contenu de l'image elle même. Parmi les moteurs de recherches d'images utilisant l'indexation basée texte, nous citons : Google (`wwww.google.com`), AltaVista photo finder (`http://image.altavista.com/`).

Toutefois, lorsque les mots environnants l'image sont ambigus ou même non pertinents au contenu de l'image, la recherche basée sur le texte seulement retournera de nombreuses images indésirables. Ainsi, les chercheurs ont recours à l'indexation basée contenu (*Content-Based Image Retrieval CBIR*) [37, 38], pour choisir des mots-clés et des descripteurs plus appropriés à l'image.

Indexation basée contenu d'image : chaque image de la collection est décrite par des descripteurs soit de bas niveau primitives tels que la texture, la forme, la couleur et l'orientation, soit par des descripteurs de haut niveau tels que des objets et personnes nommés, des émotions, etc. En revanche des descripteurs de haut niveau, les descripteurs de bas niveau n'offrent pas une information sur la sémantique de l'image. Parmi les moteurs de recherches d'images utilisant l'indexation basée contenu d'image, nous citons : Google (`http://images.google.com/`), QBIC [39], Photobook [40].

Les techniques actuelles de vision par ordinateur permettent l'extraction automatique des descripteurs de bas niveau à partir d'images, avec un bon degré d'efficacité [41]. Cependant, l'extraction des descripteurs objectifs de haut niveau à partir du contenu d'image reste un axe de recherche ouvert.

Indexation basée texte et contenu d'image : les index attribués à chaque image de la base de données peuvent être basés texte et contenu d'image à la fois. Plusieurs systèmes de recherche ont été développé sous cette catégorie, tels que Diogenes [42] et AtlasWise [43].

2.6.3 *Applications d'Indexation*

L'indexation trouve de nombreuses applications, aussi bien dans le domaine de la multimédia, que dans d'autres domaines. Nous explorons ici quelque-unes de ces applications.

Protection de la propriété intellectuelle : une des applications d'indexation concernant la propriété intellectuelle est la protection des droits d'images. Une telle application est indispensable notamment sur le Web, à cause de la facilité de faire des copies non autorisées des images et de les transmettre sur le réseau d'Internet. Un exemple de système de recherche basé contenu d'image dédié à détecter les images dupliquées sur le Web est développé par Chang et al. [44].

Application de la loi et prévention du crime : l'indexation basée contenue est utilisée pour plusieurs fins dans l'application de la loi et la prévention du crime, tels que la reconnaissance de visage, la reconnaissance de l'empreinte digitale, la correspondance de l'ADN, et dans les systèmes de surveillance. En outre, beaucoup de criminels utilisent l'Internet comme un moyen de promouvoir leurs produits et services illicites, tels que les drogues et les armes illégales. De tels sites contiennent des informations visuelles plus que de texte. L'indexation basée texte ne suffit pas donc pour les localiser. Ainsi, il est clairement indispensable de développer des outils qui aident à l'indexation basée contenu d'image pour être en mesure de localiser de tels sites.

Filtrage des contenus inappropriés : de nombreux sites Web contiennent des contenus inappropriés en termes d'éthique et politiques tels que l'appel à la violence et au racisme. Ainsi, l'indexation aide fortement à filtrer de tels sites Web.

Éducation et formation : les images peuvent être utilisées par les étudiants et par les enseignants soit comme une source d'information soit comme une illustration de leur idées. La recherche concise des images est difficile devant le grand nombre d'images disponibles sur le Web. Les systèmes de recherches basées contenu d'image aident ainsi à réduire le temps de recherche.

Recherche d'histoire et d'art : les historiens et les archéologues utilisent des données visuelles pour soutenir leur recherche. L'accès à l'œuvre d'art originale peut souvent être limité (par exemple, en raison de la distance géographique, les restrictions à la propriété, ou la condition

physique de l'œuvre). Pour contourner ce problème, les chercheurs peuvent utiliser des substituts, qui peuvent être trouvés sur le Web, sous forme de photographies ou d'images de l'œuvre. Un moteur de recherche des images Web peut donc leur faire gagner du temps dans la recherche d'un tel matériel. Des exemples de travaux récents qui tentent d'appliquer la recherche d'images de l'art et de la recherche historique comprennent ceux de Barnard et al.[45] et de Wang et al. [46].

L'indexation efficace doit alors mener à des résultats de recherche bien concis et pertinents à la requête. D'une part, les systèmes de recherche récents manquent de la sémantique, il est donc nécessaire de proposer des outils qui permettent d'extraire automatiquement la sémantique de l'image. D'autre part, la majorité des méthodes d'indexation basée contenu d'image opèrent sur des images codées sans pertes. Ces méthodes ne tolèrent aucune perte d'information lors de la compression d'image. Par contre, l'indexation est mise récemment à la disposition du grand public, où les images sont compressées avec perte. Le développement des outils permettant l'indexation des images compressées avec perte devient donc une nécessité.

Tenant compte de l'énorme quantité d'informations visuelles sur le Web, les moteurs de recherche dédiés à ce type de données sont extrêmement primitives. Tout outil qui peut aider les utilisateurs à localiser les images souhaitées dans un délai raisonnable et avec une précision acceptable, devrait ainsi être le bienvenu aux concepteurs des moteurs de recherches d'images.

2.7 CONTRAINTES ET HYPOTHÈSES DE TRAVAIL

Dans ce chapitre nous avons décrit les différentes phases de la plateforme 3D. Une diversité de modes existe : de l'acquisition à l'affichage en passant par la compression et la synthèse de vue virtuelle. Cependant, certaines contraintes sont imposées à chaque phase. Dans cette section, nous abordons ces contraintes qui seront adoptées dans la suite de cette thèse.

- Acquisition : pour les applications telles que la *3DTV* et la *FVV*, un ensemble d'images de plusieurs points de vue doit être capturé durant la phase d'acquisition pour assurer l'immersion 3D. En contrepartie, la taille des données augmente linéairement avec le nombre de vues. La représentation basée texture est ainsi un format coûteux, alors que la représentation basée profondeur facilite la transmission d'un faible nombre de textures et de profondeurs grâce à la possibilité de synthétiser des vues virtuelles au niveau du récepteur en utilisant les algorithmes de synthèse basés profondeur. Par suite, la représentation basée profondeur sera adoptée pour cette raison.
- Compression : la bande de transmission du réseau étant limitée, une compression des données 3D est une nécessité. La compression doit être efficace pour avoir la qualité optimale pour un débit donné. D'autre part, les images de profondeur dans la représentation basée profondeur possèdent des caractéristiques différentes de celles des images textures. Des approches de compression particulières à la profondeur doivent donc être proposées.
- Synthèse de vue : la qualité de la vue virtuelle synthétisée est fortement liée à la qualité de la carte de profondeur. Il est donc nécessaire d'en tenir compte pour la compression de la carte de profondeur et il est ainsi important d'évaluer la performance de la technique de compression des cartes de profondeur par l'évaluation de la qualité visuelle de la vue synthétisée.
- Affichage : la convergence des technologies 3D vers l'affichage auto-stéréoscopique et l'auto-multiscopique avec haute résolution nécessite une haute qualité visuelle des vues affichées et une faible complexité au niveau de la synthèse des vues intermédiaires.
- Fonctionnalités avancées : les applications d'interprétation du contenu des images telles que l'indexation nécessitent l'extraction de la sémantique de l'image plutôt que des informations de bas niveau.

Notre objectif est donc de proposer un schéma de codage des données 3D basées profondeur qui préserve toute la sémantique présente dans les images, tout en garantissant une efficacité de codage significatif.

Ainsi, dans un premier temps, la première partie de la thèse introduit un schéma de codage scalable incluant une méthode de compression des données de profondeur, adaptée aux caractéristiques des cartes de profondeur. La deuxième partie de la thèse développe ensuite un schéma joint de représentation fine des objets et de codage basé contenu d'images. Nous proposons ainsi une schéma d'"Autofocus 3D" couplé á un algorithme de segmentation en régions d'images 3D avec un degré de granularité fonction de l'application visée. Ceci permet d'extraire finement les objets dans la scène et de focaliser automatiquement sur une zone de profondeur.

Un tel schéma de codage permet à la fois de réduire les distorsions aux zones d'intérêts dans la carte de profondeur et dans les vues intermédiaires synthétisées, et d'extraire automatiquement des objets présents dans la scène. Ce schéma proposé s'adapte bien aux contraintes imposées par la plateforme 3D et constitue un outil efficace pour l'interprétation des contenu des images.

Le chapitre suivant présente les différentes méthodes proposées dans l'État de l'Art pour la compression des données basées profondeur.

3 MÉTHODES EXISTANTES DE CODAGE DE LA PROFONDEUR

La visualisation d'un même organisme
dans sa totalité en trois dimensions,
et éventuellement au cours du temps,
devrait aussi apporter une meilleure
appréhension de divers phénomènes
en embryologie ou en biologie cellulaire.

— Aassif Benassarou *et al.*, Visualisation 3D relief du vivant [20]

Objectifs spécifiques du chapitre :

- **Connaître** les codeurs standards des données 3D.
- **Comprendre** les propriétés des cartes de profondeur.
- **Analyser** les méthodes existantes de codage de la profondeur.

3.1 INTRODUCTION

Comme évoqué dans le chapitre précédent, les représentations d'image basées profondeur, alliant une image de texture 2D avec une carte de profondeur, sont bien adaptées pour des applications 3D récentes telles que la 3*DTV* ou la *FVV*. Cette représentation dite 2D+Z permet de transmettre un nombre réduit de vues, puis de synthétiser éventuellement au décodeur des vues manquantes.

L'image de profondeur est une image en niveau de gris qui peut donc être codée par n'importe quel outil de codage d'image classique. Toutefois, la carte de profondeur possède des caractéristiques différentes de celles des images naturelles. Il n'est ainsi pas optimal de coder la profondeur avec les outils de compression traditionnels [47]. Des algorithmes spécifiques pour la compression de la profondeur doivent ainsi être utilisés.

Dans la première partie du chapitre, nous présentons les principaux standards de codage 2D (Section 3.2), ainsi que leurs extensions 3D (Section 3.3). Dans la seconde partie, nous analysons les caractéristiques des cartes de profondeur (Section 3.4) et présentons les différents outils adaptés à leur codage (Section 3.5).

3.2 STANDARDS 2D

Les deux comités de standardisation ITU-T (VCEG) et ISO/IEC (MPEG) développaient depuis 1995 plusieurs standards de codage vidéo 2D. Les standards les plus connus dans le domaines scientifiques et industriels, sont le **MPEG-2** (finalisé en 1995), le **H.264/ Advanced Video Coding** (**H.264/AVC**) (finalisé en 2003) et récemment le **H.265/High Efficiency Video Coding** (**HEVC**) (finalisé en 2013). Ces standards furent initialement développés pour répondre aux applications telles que la téléphonie vidéo, le stockage, le broadcast, etc. Généralement, les séquences vidéos contiennent des redondances spatiales et temporelles. Par suite, la compression de ces vidéos exploitent ces redondances afin de réduire les coûts de codage. Les techniques de codage standards des vidéos 2D sont ainsi basées sur un codage hybride (prédiction/transformation). La FIGURE 33 illustre le schéma générique d'un codage vidéo hybride.

L'image d'entrée est découpée en macroblocs[1], chaque macrobloc étant composé de trois composantes Y, C_r et C_b. La luminance Y représente la luminosité. Les chrominances C_r et C_b représentent l'information de couleur. Le système visuel humain (SVH) étant moins sensible à la couleur qu'à la luminosité, les informations de chrominance sont sous-échantillonnées, généralement par un facteur 2 dans les deux directions horizontales et verticales.

1 Un macrobloc est un bloc de taille $N \times N$ pixels, avec $N > 2$

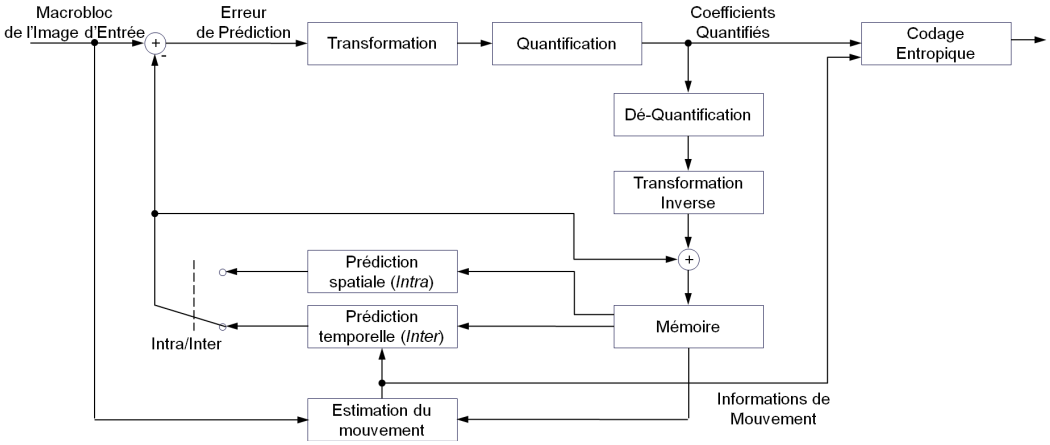


FIGURE 33. Schéma générique de codage hybride de vidéo 2D.

Ces macroblocs sont prédits soit en mode *Intra* soit en mode *Inter*. En mode *Inter*, le macrobloc est prédit par compensation de mouvement : un vecteur de mouvement est estimé et transmis pour chaque bloc. Ce vecteur représente le déplacement du bloc depuis une image déjà transmise et reconstruite en mémoire. En mode *Intra*, le macrobloc n'est plus prédit temporellement mais spatialement. L'erreur de prédiction, la différence entre le bloc orignal et le bloc prédit, est transformée, quantifiée et codée avec un codeur entropique. Afin de reconstruire la même image au décodeur, les coefficients quantifiés sont dé-quantifiés et transformés en inverse, puis ajoutés au bloc prédit. Le résultat est le macrobloc reconstruit et enregistré en mémoire.

3.2.1 *MPEG-2*

Le profil principal du standard MPEG-2 admet strictement ce schéma hybride. Les images d'entrée sont arrangées en groupes d'images (*Group Of Picture* GOP). Chaque image est découpée en macroblocs de taille fixe (16×16 pixels pour la luminance et 8×8 pixels pour chacune des chrominances). Un GOP rassemble une séquence d'images de type I (*Intra Coded Frame*), P (*Predictive Coded Frame*) et B (*Bi-directionnally Predictive Coded Frame*), comme le montre la FIGURE 34. Une image de type I est codée indépendamment des autres images de la séquence (l'information de prédiction est mise à zéro). Une image de type P est codée en mode *Inter*, elle est prédite par une compensation de mouvement (*Motion Compensation Prediction MCP*) à partir de l'image de référence (de type I) précédente uniquement. Alors qu'une image de type B est prédite par une compensation de mouvement à partir de l'image de référence précédente et de l'image de référence suivante (les images de référence sont de type P ou I), en prenant la moyenne des deux valeurs prédites.

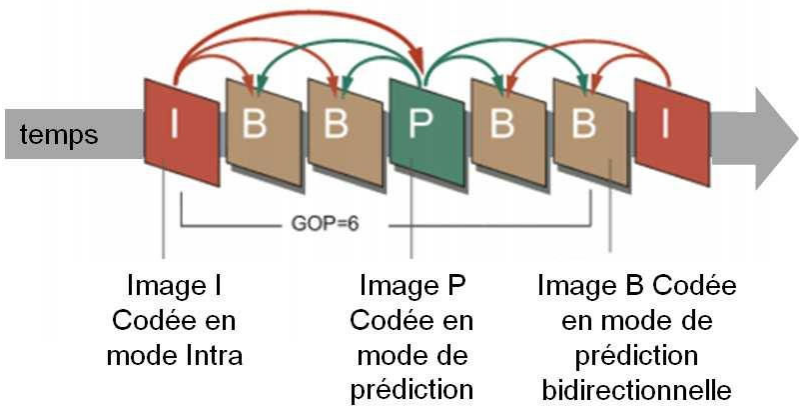


FIGURE 34. Groupe d'images GOP : dans cet exemple, le rassemblement est de type IBBPBBI.

Comme l'indique la FIGURE 35, l'encodeur détermine une fenêtre de recherche dans l'image de référence temporelle, appelée *Search Window*. L'estimation du vecteur mouvement (*Motion Vector MV*) d'un bloc courant, est la recherche de son ressemblant dans cette fenêtre (cette opération de recherche est appelée *Block Matching*). La résolution de recherche du bloc correspondant dans les images de références pour le standard MPEG-2 est de 1/2 pixel.

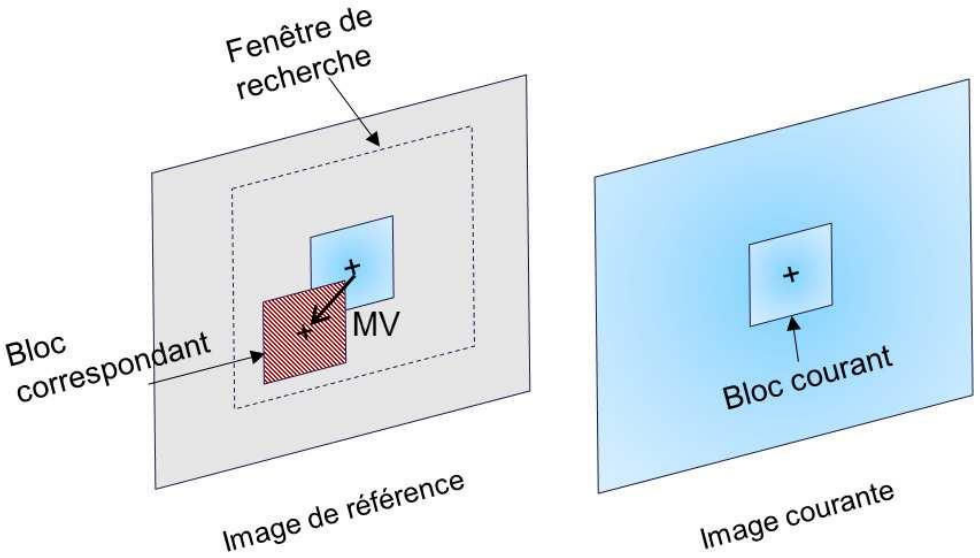


FIGURE 35. Recherche dans la fenêtre le bloc le plus ressemblant au bloc courant.

Ensuite, la transformée appliquée aux erreurs de prédiction est la transformée en cosinus discrète (*Discrete Cosine Transform DCT*). Une quantification linéaire est appliquée aux coefficients de la transformée. Les coefficients quantifiés sont ensuite codées par un codage à longueur variable *(Variable Length Code VLC)*.

Parmi les différents profils du standard MPEG-2, on s'intéresse à un profil qui est appelé "*scalable profile*", ou profil scalable en qualité. Ce profil permet le codage d'une image avec deux couches : la couche de base encode l'image à un niveau de qualité donné, puis la couche de rehaussement permet une augmentation de la qualité. Un tel profil peut être mis à profit pour la transmission de vidéos sur des réseaux hétérogènes (des réseaux supportant une multiplicité de capacités et de complexités).

3.2.2 *H.264/AVC*

H.264/AVC [51] marqua, en termes d'efficacité de compression, une réelle rupture par rapport aux standards existants. Ce schéma également hybride (FIGURE 33) introduisit de nouvelles techniques :

- L'image peut être partitionnée en tranches (*Slices*) de type I, P et B, codées indépendamment[2](voir FIGURE 36).
- Afin de réduire les effets de blocs induits par la partition de l'image en macroblocs, un filtre adaptatif appelé *Deblocking filter*, est utilisé dans la boucle de prédiction. Le macrobloc traité est enregistré en mémoire et peut être utilisé pour la prédiction des futurs macroblocs.
- Alors que dans le standard MPEG-2, la prédiction *Inter* utilise une seule image de référence précédente, le standard H.264/AVC permet la prédiction à partir de plusieurs références temporelles précédentes (*Multiple reference frames*) (voir FIGURE 37). Le standard H.264/AVC permet ensuite une combinaison linéaire des valeurs prédites.

2 Le but du partitionnement en slices est la re-synchronisation dans le cas de pertes de données lors de transmission.

- L'estimation du vecteur de mouvement est de plus haute précision que celle dans le standard MPEG-2 (précision au 1/4 pixel).
- Dans le mode de prédiction *Intra* du standard H.264/AVC, une prédiction spatiale est utilisée. Elle utilise les macroblocs déjà reconstruits de la même image pour la prédiction du macrobloc courant (voir FIGURE 38).
- La transformée DCT est approximée par une transformée entière.

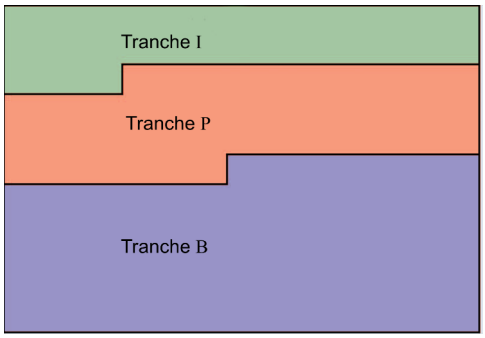


FIGURE 36. Partition de l'image en plusieurs tranches.

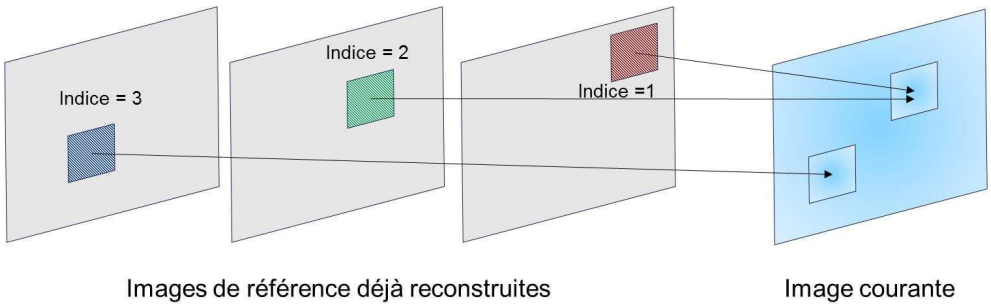
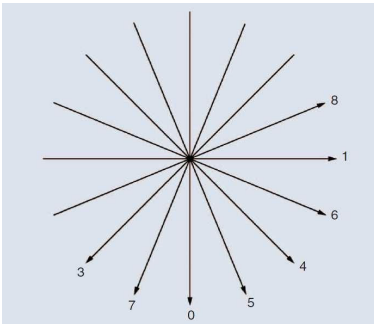


FIGURE 37. Prédiction par compensation de mouvement à partir de plusieurs images de référence précédentes. En plus de l'information du vecteur mouvement, l'indice de l'image référence doit être transmis.

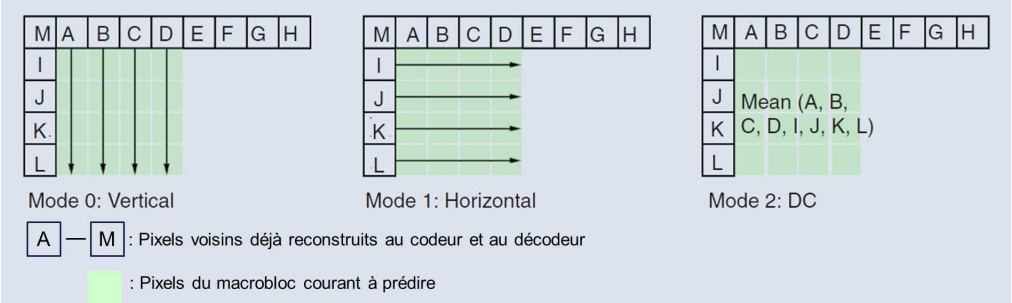
3.2.3 *HEVC*

Le standard HEVC fut développé en focalisant plus particulièrement sur deux points : l'augmentation de la résolution de la vidéo et l'augmentation de l'utilisation des architectures de traitements parallèles. Toujours basé sur le concept de codage hybride, le HEVC apporte plusieurs modifications par rapport aux standards antérieurs. L'ensemble de ces modifications permet au standard HEVC de coder une séquence d'images avec un gain de débit d'environ 50% par rapport aux standards antérieurs pour une même qualité visuelle. Parmi les modifications essentielles :

- Un partitionnement de l'image en blocs de taille variable ($64 \times 64, 32 \times 32$, jusqu'à 8×8 pixels) (voir FIGURE 39).
- Nombre d'orientations possibles plus important pour le mode de prédiction *Intra* (voir FIGURE 40).
- L'introduction d'un mode MERGE, dans lequel le bloc courant hérite des informations de prédiction d'un candidat donné, sélectionné parmi les voisins spatiaux ou temporels.



(a) 8 directions possibles de prédiction Intra dans le standard H.264/AVC.



(b) 3 modes de prédiction *Intra* parmi 9 : Par exemple, si la prédiction verticale (mode 0) est appliquée, tous les pixels en dessous de A sont prédits à partir de A, tous les pixels en dessous de B sont prédits à partir de B, et ainsi de suite.

FIGURE 38. Prédiction Intra du standard H.264/AVC

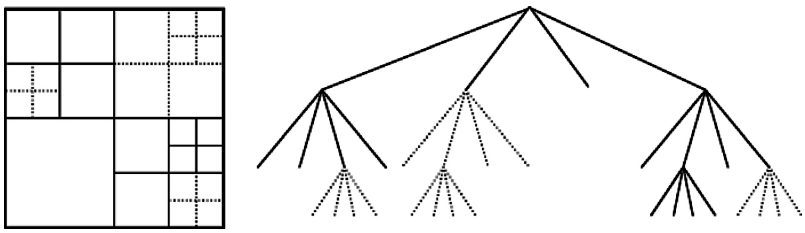


FIGURE 39. Partition de l'image en blocs de taille variable avec l'arbre quaternaire *QuadTree* correspondant.

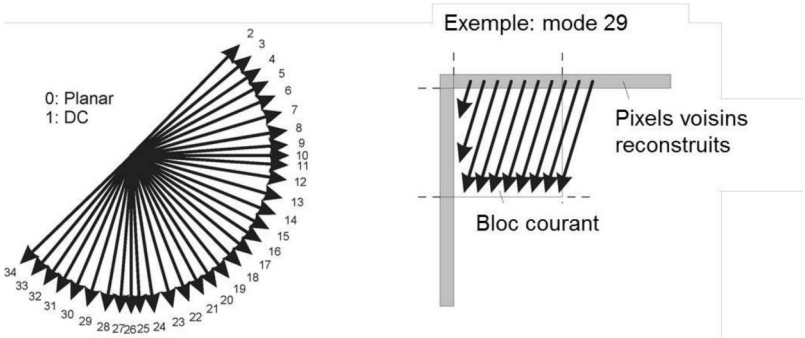


FIGURE 40. 33 directions possibles de prédiction Intra dans HEVC.

Basées sur les standards 2D décrits dans la Section 3.2, des extensions au codage 3D furent développées par le groupe JVT-3V. Nous présentons par la suite les différentes extensions des standards aux formats 3D.

3.3.1 *Codages standards du couple Stéréo*

Le standard **MPEG-2 MultiView Profile (MPEG-2 MVP)**, extension du profil scalable du standard MPEG-2, fut le premier standard proposé pour exploiter la corrélation inter-vue entre les deux vues du couple stéréoscopique. La première vue (vue gauche par exemple) est considérée comme couche de base (*base layer*), et la seconde comme couche de rehaussement (*enhancement layer*). Dans un premier temps, la séquence vidéo de la couche de base est codée avec les outils de prédiction du standard MPEG-2 (voir FIGURE 41).

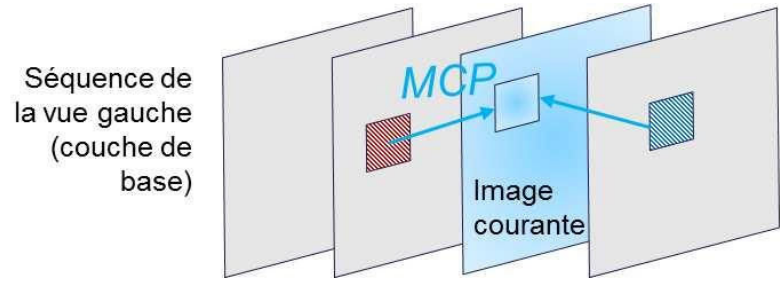


FIGURE 41. Prédiction par compensation du mouvement (MCP) de la couche de base dans MPEG-2 MVP.

La seconde vue est ensuite codée. Les images de la couche de rehaussement (vue droite dans ce cas là) utilisent alors la prédiction temporelle *MCP* uni-directionnelle, c.à.d réalisée à partir de l'image précédente uniquement (voir FIGURE 42). En plus de la prédiction temporelle, le standard **MPEG-2 MVP** profite de la corrélation spatiale entre les deux vues en appliquant une prédiction inter-vues (entre les 2 vues adjacentes). Cette opération est appelée prédiction par compensation de disparité (*Disparity-Compensated Predcition* DCP) (voir FIGURE 42).

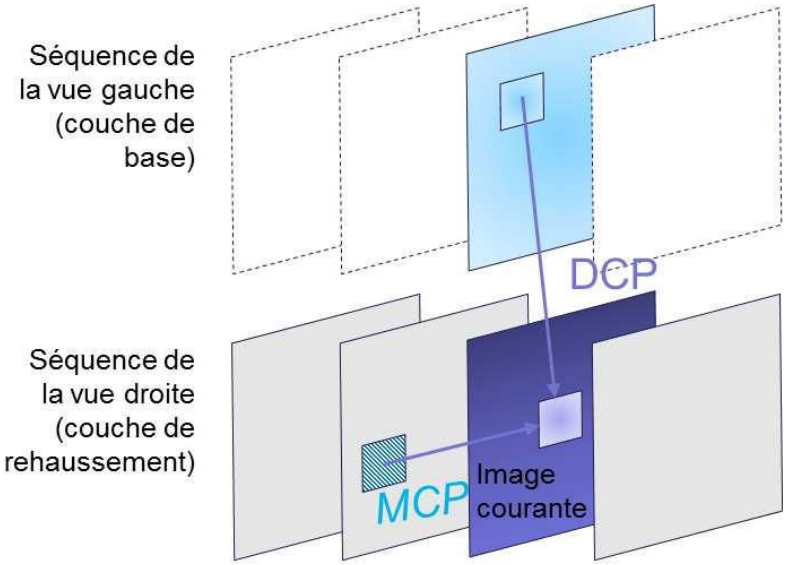


FIGURE 42. Prédiction par compensation de disparité (DCP) de la couche de rehaussement dans MPEG-2 MVP.

La DCP consiste à chercher à l'instant courant, dans l'image de la vue adjacente, le macrobloc le plus ressemblant à celui de l'image courante. Les macroblocs correspondants sont identifiés par un vecteur de disparité (*Diparity Vector* DV). Finalement, les paramètres des caméras (focale,

ligne de base...) sont également encodés dans le flux de la couche de rehaussement. Le flux binaire résultant peut être décodé par les décodeurs 2D de MPEG-2 (retro-compatibilité). Un exemple de codage MPEG-2 MVP est donné (voir FIGURE 43).

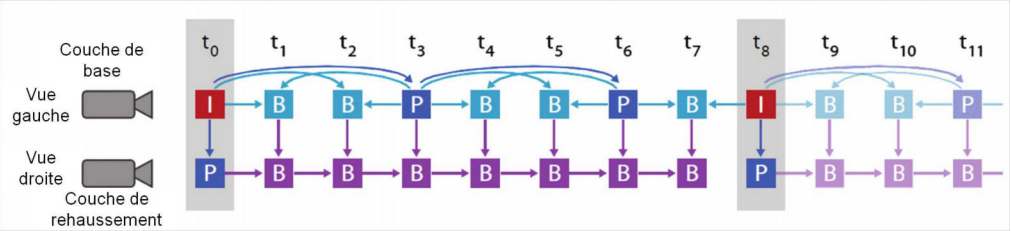


FIGURE 43. Schéma de MPEG-2 Multiview Profile avec un GOP composé de "IBBP" dans cet exemple : vue de gauche considérée comme couche de base.

Le standard **H.264/AVC stereo SEI message** est une extension du standard H.264/AVC, qui fut proposé pour le codage des formats *Frame compatible (FC)* expliqué en Chapitre 2. Il utilise une information auxiliaire (*Supplementary Enhancement Information SEI*), pour que les échantillons soient correctement interprétés et désentrelacés au niveau du décodeur.

3.3.2 *Codages standards des données 2D+Z*

En 2007, le **MPEG-C Part 3** fut la première norme dédiée au codage des données 3D de type 2D plus profondeur 2D+Z. Elle permet l'ajout d'un flux auxiliaire associé au flux vidéo standard qui peut être interprété par le décodeur. Les deux flux sont codés indépendamment avec H.264/AVC afin de produire deux flux binaires distincts qui seront ensuite rassemblés par entrelacement temporel. Ce flux auxiliaire peut donc contenir l'information de profondeur (voir FIGURE 44). Étant une extension du standard H.264/AVC, ce codage reste retro-compatible avec les décodeurs existants. Cependant, la compression de la carte de profondeur se fait toujours par des standards dédiées aux images couleurs naturelles (H.264/AVC). Par suite, la compression n'a pas été adaptée aux caractéristiques de la carte de profondeur, d'où la nécessité de proposer des schémas de codage de la carte de profondeur qui répondent mieux aux caractéristiques de celle ci.

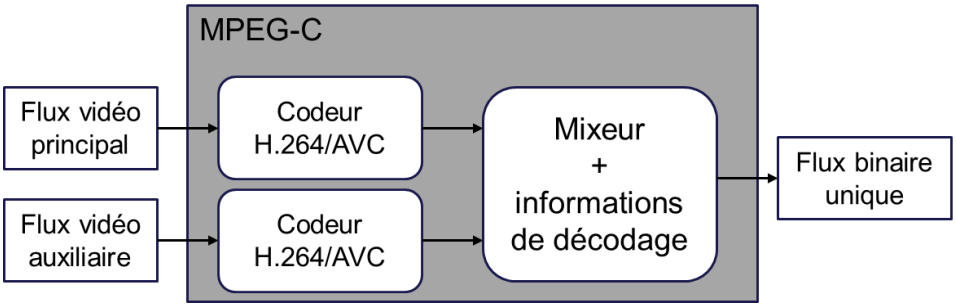


FIGURE 44. Encodage d'un flux 2D+Z via MPEG-C part 3

3.3.3 *Codages standards de MVV*

En 2010, le groupe JCT-3V a finalisé le standard dédié au format MVV, nommé **H.264/ Multiview Video Coding (H.264/MVC)**. Il s'agit d'une extension multivues du codeur 2D standard H.264/AVC. Le H.264/MVC se base sur les principes de codage du standard H.264/AVC, tels que le rassemblement en GOP, la prédiction spatiale dans l'image elle même (prédiction *Intra*) et la prédiction temporelle avec références multiples (prédiction *Inter*) (expliquée dans la sous-section 3.2.2). La première vue est considérée comme une vue de base et elle est codée indépendamment par le standard H.264/AVC. Les autres vues sont codées avec des prédictions spatiales, temporelles, et encore avec une prédiction inter-vue à partir des images correspon-

dantes des vues adjacentes (*Disaprity-Compensated Prediction DCP*) [48] (voir FIGURE 45). Le principe de la DCP est identique à celui utilisé dans MPEG-2 MVP (voir la sous-section 3.3.1). Deux configurations sont possibles :

- configuration *view progressive* : pour une vue déterminée, seule la première image d'un GOP est codée avec la prédiction inter-vue, les autres images sont codées avec une prédiction temporelle,

- configuration *fully hierarchical* : pour une vue déterminée, les prédictions inter-vues bi-directionnelles sont permises pour toutes les images du GOP.

Au niveau du décodeur, la vue de base peut être décodée indépendamment des autre vues.

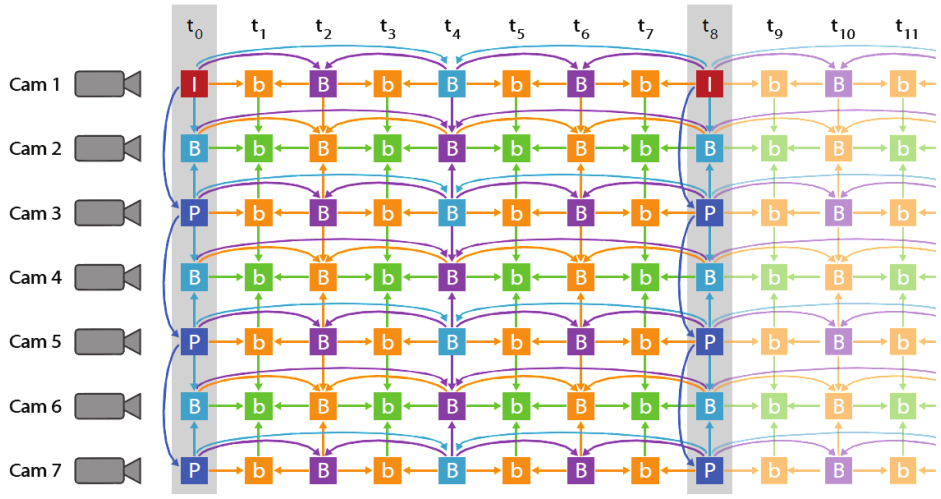


FIGURE 45. Encodage par H.264/MVC de 7 vues. La vue de base est la vue 1. Les vues de 2 à 6 sont codées en *fully hierarchical*. La vue 7 est codée en utilisant la prédiction inter-vues seulement pour la première image du GOP (*view progressive*).

De nombreux travaux de recherche se focalisent sur l'optimisation des modes de prédiction inter-vues dans H.264/MVC. Certains cherchent à optimiser les algorithmes de prédiction inter-vues afin de réduire leur complexité [49], ou pour améliorer la prédiction [50, 51, 52]. D'autres travaux proposent de nouveaux algorithmes pour la prédiction inter-vues afin d'améliorer l'estimation des vecteurs de disparité (*Disparity Vector*), et ça avec une faible complexité [53].

Récemment, une extension **MV-HEVC** du nouveau standard HEVC au codage des données MVV a été introduite [14, 15]. Le concept est similaire à celui de l'extension multivues de H.264/AVC : une vue de base est codée avec le standard HEVC, les autres vues sont codées avec les outils du standard HEVC et en plus avec une prédiction inter-vue. Ce standard tire profit des performances d'HEVC par rapport à H.264/AVC, pour un codage multivues très efficace.

3.3.4 *Codages standards de MVD*

Actuellement, le groupe JCT-3V finalise deux nouveaux standards pour le codage des données multivues plus profondeur ou MVD. Le premier est une extension multivues plus profondeur compatible avec le standard H.264/AVC. Le second est compatible avec le standard HEVC. Nous détaillons ces deux techniques dans les paragraphes suivants.

Pour la première technique, deux approches sont proposées. La première, appelée **MVC-plus-depth (MVC+D)**, fut introduite en 2013 comme une extension retro-compatible de H.264/MVC. La séquence vidéo (texture) des différentes vues est tout d'abord encodée par un codeur H.264/MVC (voir sous-section 3.3.3). Puis dans un flux séparé, les séquences de profondeur associées aux différentes vues sont également encodées par un codeur H.264/MVC, mais indépendamment de la texture, tout en exploitant les prédictions temporelles et inter-vues entre les cartes de profondeur (voir FIGURE 46).

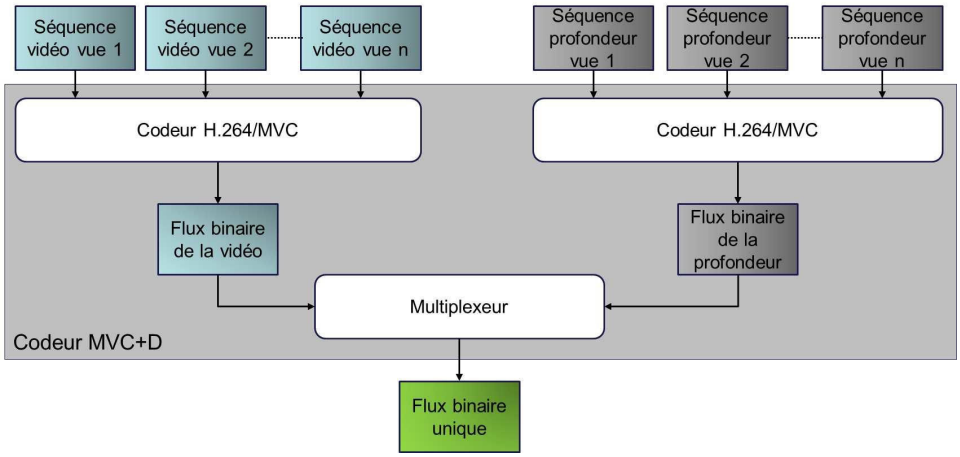


FIGURE 46. MVC + D : Extension de H.264/MVC.

Une seconde approche fut proposée fin 2013, appelée 3**D-AVC**, comme extension de H.264/AVC pour un codage joint multivues plus profondeur. En se basant également sur le standard H.264/AVC, le 3D-AVC encode cette fois-ci la profondeur et la texture de manière dépendante. Différents outils de codage de la texture et de la profondeur adaptés ont été développés à cette fin. Pour le codage de la texture, les auteurs proposent dans [54] une prédiction basée profondeur des vecteurs mouvements de la texture, exploitant les redondances entre la texture et la profondeur. Pour le codage de la profondeur, un filtrage joint entre les différentes vues des cartes de profondeur est proposé dans [55], afin d'augmenter la fidélité de l'information de profondeur entre les vues. Dans [56], le processus de prédiction inter-vues de la profondeur est pondéré par les valeurs extrêmes des plans de profondeur Z_{proche} et Z_{loin}. Les outils de codage de la profondeur seront détaillés dans la Section 3.5.

La seconde classe de méthodes qui se développe actuellement au sein du groupe JCT-3V repose sur une extension multivues plus profondeur du standard HEVC, appelée 3**D-HEVC**. Ces extensions 3D [57] (voir FIGURE 47) apportées au standard 2D peuvent être résumées comme suit :

- Codage des séquences vidéos des différentes vues à travers une prédiction par compensation de disparité (*Disparity Compensated Prediction DCP*), identique à celle proposée dans MV-HEVC et H.264/MVC, et une prédiction du mouvement et de l'erreur résiduelle en se basant sur les vues adjacentes (*Inter-view Motion Prediction* et *Inter-view Residual Prediction*).
- Codage des cartes de profondeur en utilisant de nouveaux modes de prédiction *Intra*, une prédiction par compensation de mouvement modifiée, une prédiction par compensation de disparité (DCP) et héritage des vecteurs de mouvements.
- Encodage basé sur l'optimisation des vues synthétisées.

Les deux derniers outils seront détaillés dans la Section 3.5.
La FIGURE 48 illustre l'évolution des codeurs standards 2D et 3D.

Les cartes de profondeur ayant des caractéristiques différentes de l'image couleur, une étude détaillée de ses caractéristiques doit être effectuée afin de pouvoir fournir un codeur efficace de la carte de profondeur. Cette étude est présentée dans les sections suivantes.

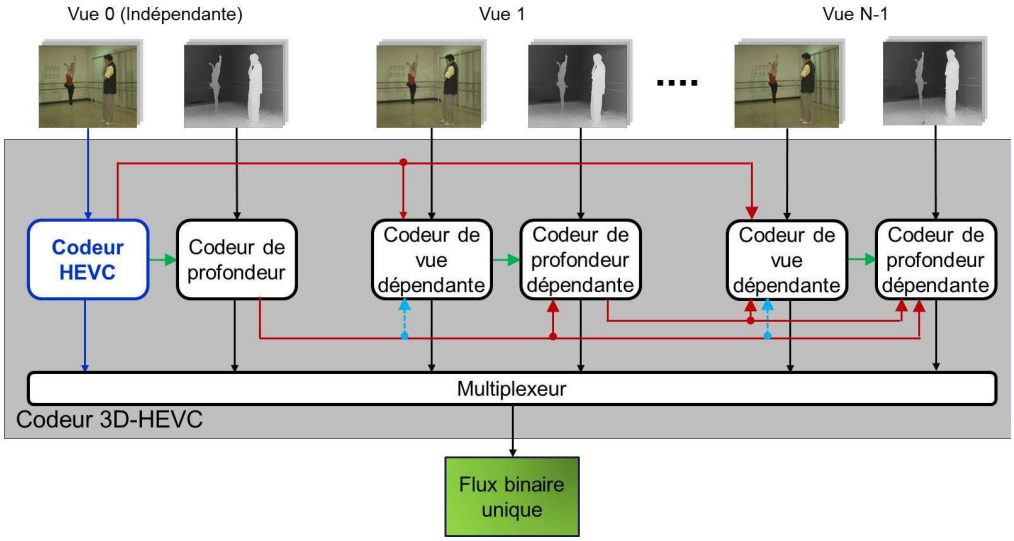
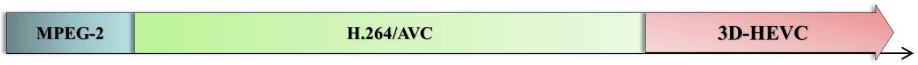
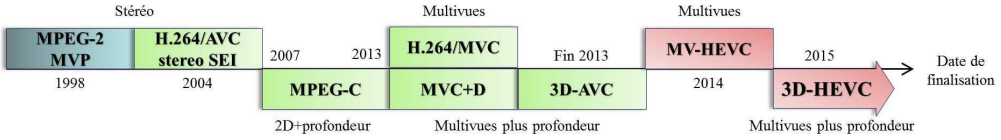


FIGURE 47. Extension 3D du standard HEVC (3D-HEVC).



(a) Standards Video 2D



(b) Standards Vidéo 3D

FIGURE 48. Évolution des codeurs standards (a) 2D et (b) 3D.

3.4 CARACTÉRISTIQUES D'UNE CARTE DE PROFONDEUR

Chaque objet dans la scène est situé à une certaine distance de la caméra (nommée profondeur réelle) entre le plan de profondeur le plus proche Z_{proche} et le plus éloigné Z_{loin}. Ainsi, à chaque pixel de l'image texture, une valeur est associée en niveau de gris comprise entre 0 (plus loin) et 255 (plus proche) comme illustré dans la FIGURE 49. La carte de profondeur est donc vue comme une image de luminance. Au contraire des images naturelles, une carte de profondeur n'est pas affectée par l'illumination et ne contient ni ombre ni texture. Elle est de plus caractérisée par de larges régions lisses séparées par des contours nets [47].

FIGURE 49. Image de texture et carte de profondeur associée.

3.5 MÉTHODES DE CODAGE DE LA CARTE DE PROFONDEUR

Plusieurs outils et méthodes de compression de la carte de profondeur ont été proposés. Selon *Lucas et al.* dans [20], ils peuvent être groupés sous 3 catégories : 1) les outils se basant sur les caractéristiques intrinsèques de la carte de profondeur, 2) ceux exploitant les corrélations avec la texture associée et enfin, 3) ceux optimisant la compression de la carte de profondeur pour la qualité des vues intermédiaires synthétisées.

3.5.1 *Méthodes exploitant les caractéristiques intrinsèques des cartes de profondeur*

Ces méthodes peuvent être divisées en deux sous catégories : les outils qui traitent la carte de profondeur dans son ensemble (haut niveau) et ceux qui opèrent au niveau des blocs.

3.5.1.1 *Outils de codage de haut niveau*

Un premier exemple d'outil de cette sous-catégorie se base sur le principe que la carte de profondeur peut être codée à une résolution spatiale plus faible que celle de la texture, tout en conservant à peu près la même qualité des vues synthétisées. Ainsi, cet outil consiste à réduire la résolution de la carte de profondeur avant codage par un sous-échantillonage suivi d'un sur-échantillonage. Par exemple, le standard MVC+D permet le codage des images texture et profondeur à résolutions asymétriques (voir FIGURE 50). Dans le modèle de test pour les solutions 3DV (*3DV Test Model*) [56], la résolution spatiale de la carte de profondeur est réduite au quart de la résolution initiale (la moitié dans chaque direction, $m = n = 0.5$).

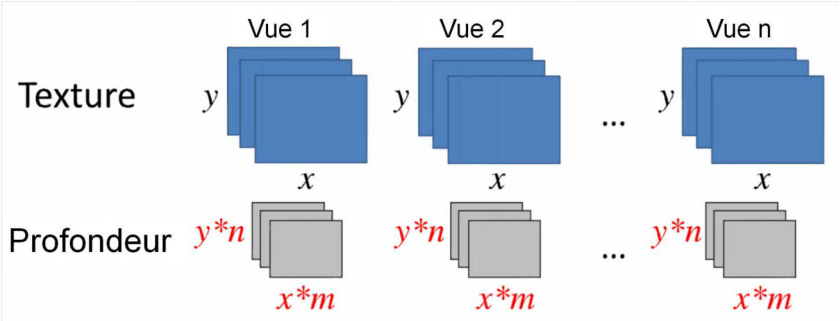


FIGURE 50. Exemple de résolution des images texture et profondeur, avec m et $n \in [0, 1]$.

Cet outil entraîne une réduction de la taille des données à coder et apporte un gain de compression important. Néanmoins, la réduction de la résolution spatiale de la carte de profondeur implique une réduction de la qualité, notamment aux alentours des contours. Ceci va causer des artéfacts importants dans les vues synthétisées. Une approche de sur-échantillonnage non linéaire de la carte de profondeur est proposée dans [58], afin de préserver les contours des objets dans la carte de profondeur, et par suite améliorer la qualité objective et visuelle.

La réduction de la résolution des vecteurs mouvements (*Motion Vector MV*) et des vecteurs disparités (*Disparity Vector DV*) utilisées lors du codage de la profondeur est un autre outil de codage de la profondeur. Par exemple, lors du codage de la texture par 3DVC, la résolution du MV (DV) est de 1/4 du pixel, alors que lors du codage de la profondeur, elle est réduite à 1 pixel [59]. Cette réduction de précision des vecteurs mouvements et disparités réduit à son tour le débit généré.

Un autre exemple d'outil de codage performant est la prédiction de la carte de profondeur par synthèse de vues (*View Synthesis Prediciton* VSP). En utilisant un algorithme de synthèse de vue (tel que le *DIBR*, voir Section 2.4), et suivant les paramètres des caméras, une carte de profondeur, déjà reconstruite d'un point de vue adjacent, peut être projetée vers le plan du point de vue courant à coder. La carte synthétisée peut servir comme prédicteur de la carte courante à coder. Cet outil est introduit dans [18, 60] et implémenté dans le 3DV Test Model [56].

Puisque chaque carte de profondeur peut être créée avec différentes valeurs extrémales de profondeur[3] (Z_{proche} et Z_{loin}), une valeur de profondeur réelle Z peut être représentée, après échelonnage entre 0 et 255, avec différents valeurs en niveau de gris D. Cela conduit à une mauvaise prédiction temporelle ou inter-vues et réduit donc les gains de codage. Deux solutions étaient proposées dans le 3DV Test Model [56]. Une première solution proposée dans le 3DV Test Model, pour augmenter la cohérence entre les différentes vues de la carte de profondeur, est le filtrage joint des cartes de profondeur avant leur encodage (*Joint View Depth Filtering* JVDF) [55]. Cette solution est divisée en 3 étapes essentielles : 1) Les cartes de profondeur de toutes les N vues disponibles sont projetées vers le même point de vue *m* (voir FIGURE 51). En supposant que les N caméras sont arrangées d'une manière parallèle, la projection d'un pixel de coordonnées (x_n, y_n) d'une vue n vers la vue m consiste en un déplacement horizontal calculé par l'équation (6).

$$\begin{aligned} y_m &= y_n \\ x_m &= x_n + d = x_n + f.l_{m,n}/Z \end{aligned} \tag{6}$$

où d est la disparité entre les deux vues m et n, calculée en fonction de la distance focale f de la caméra, la ligne de base $l_{m,n}$ entre les deux caméras M et N et la profondeur réelle Z (comme expliqué dans le Chapitre 2). À l'issue de cette première étape, $N-1$ cartes de profondeur estimées sont générées pour le point de vue m. 2) Ensuite, un filtrage ou "fusion" pondérée des cartes estimées Z_i est appliqué pour produire une carte de profondeur "sans bruit" (*noise-free*) $\hat{Z}_m$, suivant l'équation (7).

$$\hat{Z}_m(x,y) = w_m.Z_m(x,y) + \sum_{i=0}^{N-1} w_i.Z_i(x,y) \tag{7}$$

$$\text{avec } w_i = \begin{cases} 1, & si\ \frac{|Z_i(x,y)-Z_m(x,y)|}{Z_m(x,y)} < T \\ 0, & \textit{autrement} \end{cases}$$

où T est le seuil qui contrôle le degré du filtrage qui peut être déterminé automatiquement ou par l'utilisateur. 3) Enfin, la carte estimée $\hat{Z}_m$ est re-projetée vers les $N-1$ points de vue originales.

Cette solution est presque similaire à celles proposées dans [61, 62], mais elle est moins complexe.

3 Différentes trames d'une même vue ou différentes vues à un même instant, d'une séquence vidéo de profondeur, peuvent avoir différentes valeurs extrémales de profondeur[20].

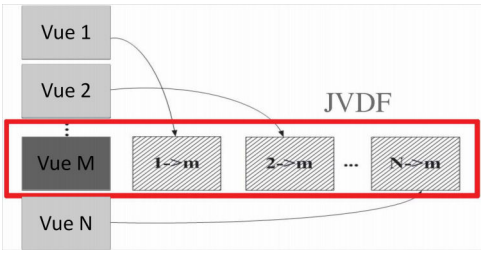


FIGURE 51. Concept global de *JVDF*.

La deuxième solution de compensation de la différence des valeurs extrémales entre les différentes vues est dénotée *Depth-Range Weighted Prediction* DRWP [56]. Elle consiste à pondérer l'opération de prédiction dans le codeur H.264/AVC, en fonction des valeurs extrémales des cartes de profondeurs de référence et celle à coder, comme suit :

$$v_2 = \lfloor v_1.W + \mathrm{Offset} + 0.5 \rfloor \tag{8}$$

$$W = \frac{Z_{loin1} - Z_{proche1}}{Z_{loin2} - Z_{proche2}} \times \frac{Z_{loin2} \times Z_{proche2}}{Z_{loin1} \times Z_{proche1}} \tag{9}$$

$$\mathrm{Offset} = \frac{255 \times Z_{loin2}}{Z_{loin1}} \times \frac{Z_{loin2} - Z_{loin1}}{Z_{loin2} - Z_{proche2}} \tag{10}$$

où les variables avec l'indice 1 représentent les paramètres de la carte de profondeur courante à coder et les variables avec l'indice 2 représentent les paramètres de la carte de profondeur référence.

3.5.1.2 *Outils de codage par blocs*

Le premier outil à être présenté dans cette catégorie est le partitionnement non rectangulaire des cartes de profondeur. Cette représentation en forme arbitraire permet la modélisation des régions de la carte de profondeur par des fonctions constantes ou linéaires, séparées par une ligne droite tout au long de leur limites. Ceci a été présenté dans [63]. Les régions lisses de la carte de profondeur sont approximées par une fonction constante par morceaux alors que les régions avec une variation graduelle de la profondeur (i.e. mur ou sol) sont approximées par une fonction linéaire par morceaux. Le bloc qui ne peut être modélisé par aucune de ces fonctions sera divisé en 4 sous-blocs (arbre quaternaire *Quadtree*). Le processus est ensuite itéré pour chaque bloc jusqu'à ce que toutes les feuilles de l'arbre quaternaire soient approximées par une fonction de modélisation (voir FIGURE 52).

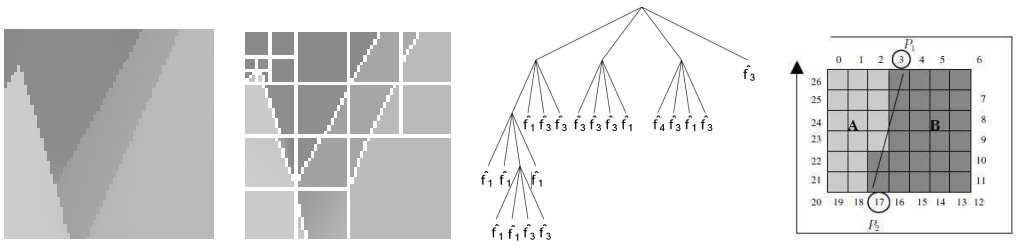


FIGURE 52. Exemple de décomposition *Quadtree*. Chaque bloc (b), représenté par un nœud dans le *QuadTree* (c), est approximé par une fonction de modélisation [64].

Les quatre fonctions de modélisations proposées sont les suivantes :

- $\hat{f}_1$: fonction constante.
- $\hat{f}_2$: fonction linéaire.

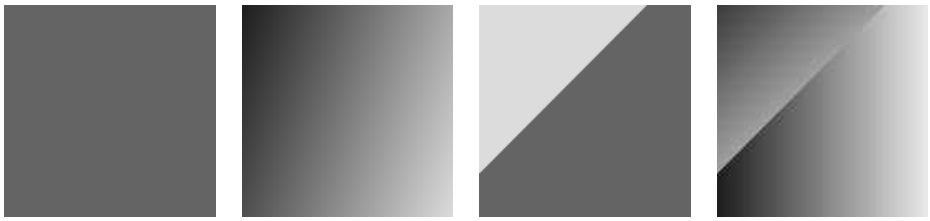
- $\hat{f}_3$: fonction constante par morceaux (*wedgelet*). Le bloc est modélisé par deux régions A et B séparées par une ligne droite et approximées chacune par une fonction constante (voir FIGURE 52).

$$\hat{f}_3(x,y) = \begin{cases} \hat{f}_{3A}(x,y) & = \gamma_{0A} \quad (x,y) \in A \\ \hat{f}_{3B}(x,y) & = \gamma_{0B} \quad (x,y) \in B \end{cases}$$

- $\hat{f}_4$: fonction linéaire par morceaux (*platelet*). Le bloc est modélisé par deux régions A et B séparées par une ligne droite et approximées chacune par une fonction linéaire (voir FIGURE 52).

$$\hat{f}_4(x,y) = \begin{cases} \hat{f}_{4A}(x,y) & = \theta_{0A} + \theta_{1A}x + \theta_{2A}y \quad (x,y) \in A \\ \hat{f}_{4B}(x,y) & = \theta_{0B} + \theta_{1B}x + \theta_{2B}y \quad (x,y) \in B \end{cases}$$

La FIGURE 53 illustre des exemples motifs des fonctions de modélisations. Pour chaque bloc, les meilleurs coefficients des fonctions de modélisation sont cherchés par optimisation débit-distorsion. Ainsi, les informations envoyées au décodeur sont les coefficients des fonctions quantifiés accompagnés de l'information de partitionnement *QuadTree*.



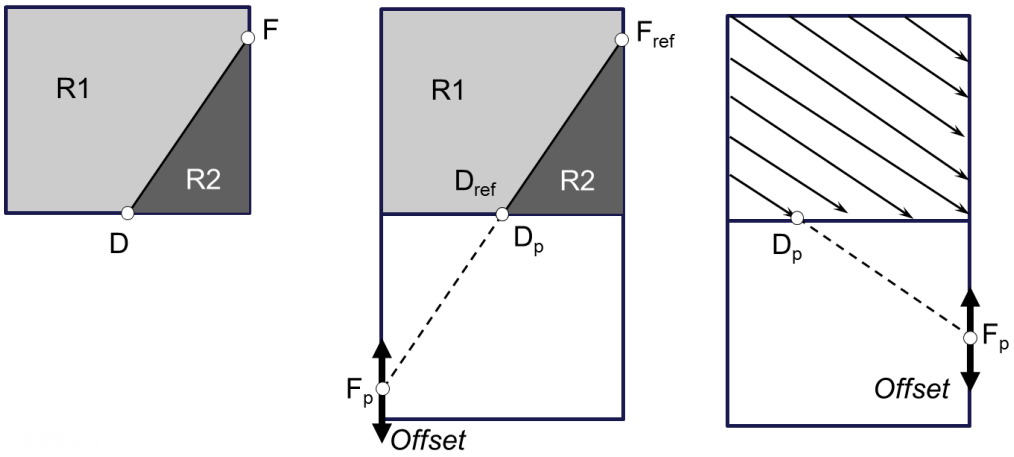
(a) $\hat{f}_1$ (b) $\hat{f}_2$ (c) $\hat{f}_3$ (d) $\hat{f}_4$

FIGURE 53. Exemple de motifs des fonctions de modélisation $\hat{f}_1$, $\hat{f}_2$, $\hat{f}_3$, $\hat{f}_4$ [64].

L'outil de modélisation de la carte de profondeur par un *wedgelet* est repris et amélioré dans le 3D-HEVC. Comme mentionné dans la sous-section 3.3.4, de nouveaux modes de prédiction *Intra*, dénotés *Depth Modeling Modes* DMM, sont ajoutés à 3D-HEVC. Dans le mode 1, le bloc de profondeur courant est approximé par 2 régions constantes, notées R1 et R2, séparées par un segment de droite DF (voir FIGURE 54a). La valeur constante P_i de la région R_i, avec $i = \{1;2\}$, est égale à la moyenne des valeurs des pixels du bloc courant recouvert par R_i. Ces deux valeurs sont prédites à partir des blocs voisins au bloc courant. La différence entre la valeur moyenne originale du bloc courant et la valeur prédite, constitue l'erreur résiduelle. Ainsi, pour ce mode, les informations envoyées au décodeur sont les erreurs résiduelles quantifiées de P_1 et P_2 et la position des points de début et de fin (D et F) du segment séparant les deux régions.

Dans le mode 2, les erreurs résiduelles de prédiction de P_1 et P_2 sont transmises mais pas la position du segment DF. Celle ci peut être prédite du bloc voisin en haut du bloc courant. Si le bloc voisin de référence spatial a déjà utilisé le mode 1 ou 2, l'idée de prédiction consiste juste de prolonger le segment $D_{ref}F_{ref}$ du bloc de référence spatial. Si le bloc de référence spatial utilise un mode *Intra* classique, le point de début D est déduit des blocs à gauche et en haut du bloc courant. Ensuite, la direction du segment DF est déterminé suivant la direction de la prédiction *Intra* du bloc référence spatial. Toutefois, le prolongement résultant peut ne pas être adéquat pour le bloc courant. Ainsi, un *offset* qui corrige le point d'arrivée F de la ligne droite, comme indiqué sur la FIGURE 54b est transmis également au décodeur.

Alors que les deux premiers outils étaient destinés à la prédiction *Intra*, un outil modifié d'estimation du vecteur mouvement MV de la carte de profondeur, proposé dans [64], concerne la prédiction *Inter*. La façon classique d'estimer le vecteur mouvement d'un bloc courant est de rechercher le bloc le plus ressemblant dans une carte de référence temporelle (*Block Matching*). Le principe est illustré dans la FIGURE 55. Une fenêtre de recherche, dénotée *Search Window*, est déterminée tout d'abord à l'intérieur des images de référence temporelle et délimitée par



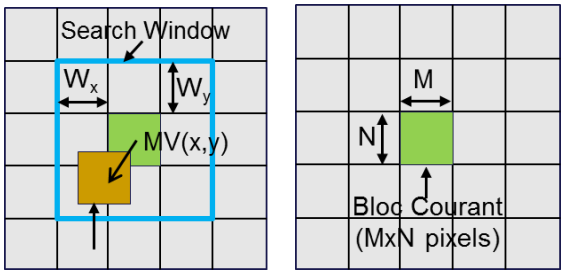
(a) DMM mode 1 (b) DMM mode 2

FIGURE 54. Modes de modélisations de profondeur 1 et 2 dans 3D-HEVC [21].

un maximum de translation horizontale (W_x) et verticale (W_y). Le meilleur bloc candidat choisi à l'intérieur de cette fenêtre est celui qui minimise la somme des erreurs quadratiques (*Sum Squared Errors* SSE). La SSE est définie comme suit :

$$SSE(i,j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (f(m,n) - g(m+i,nj))^2 \tag{11}$$

où f est le bloc courant orignal de taille MxN à coder, et g est le bloc de référence temporel de même taille.



(a) Carte de profondeur courante (b) Carte de profondeur de référence temporelle

FIGURE 55. Recherche dans l'espace x et y du bloc de référence temporel.

Cette recherche classique est effectuée suivant les deux dimensions horizontale et verticale. Cependant, le déplacement dans un espace 3D peut supposer en plus un déplacement dans la dimension de la profondeur. Ainsi, afin d'augmenter la précision de la recherche du bloc de référence temporel, *Kamolrat et al.* [64] proposent d'étendre la fenêtre de recherche à la dimension profondeur *3D_Block Matching*(voir FIGURE 56). Le calcul de la SSE est redéfini comme suit :

$$SSE(i,j,k) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (f(m,n) - g(m+i,nj) + k)^2 \tag{12}$$

où la fenêtre de recherche dans la dimension de profondeur est définie par $k_{min} \leqslant k \leqslant k_{max}$, avec k est la distance de déplacement dans la direction de profondeur. Ainsi, à la position optimale où $i = x$, $j = y$ et $k = z$, le vecteur mouvement $MV(x,y,z)$ représente la translation dans les directions horizontale, verticale et de profondeur du bloc courant.

À bas débit, le *2D-Block Matching* s'avère plus efficace que le *3D-Block Matching* du fait des bits supplémentaires nécessaires pour le codage de la troisième dimension (composante z) du vecteur

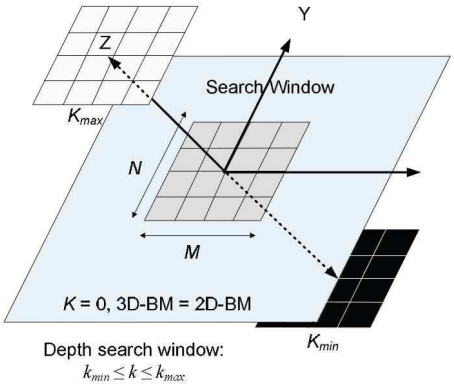


FIGURE 56. Recherche dans les espaces x, y et z du bloc de référence temporel.

mouvement. Inversement, à débit élevé, les gains deviennent prépondérants. Autre solution est proposée dans [65], qui consiste à réaliser une sélection adaptative *2D-Block Matching\ 3D-Block Matching* par optimisation débit-distorsion au niveau de chaque bloc. Le choix est ensuite signalé au décodeur pour que la solution soit décodable.

Un dernier exemple que nous présentons d'outil de codage par bloc de la carte de profondeur est celui basé sur le codage à haute qualité des contours dans la carte de profondeur, afin d'assurer une bonne qualité de synthèse de vue. Dans [66], les auteurs développent ainsi un outil de codage sans perte des contours (*Lossless Edge Coding*) incluant 3 étapes au niveau du codeur : 1) détecter soigneusement les contours des objets dans la carte de profondeur en utilisant un filtre Sobel, 2) encoder sans perte les positions des contours, 3) encoder sans perte les valeurs de luminance des deux cotés du contour. Au niveau du décodeur, la carte de profondeur reconstruite ne contient que les contours et les valeurs de profondeur des deux cotés de chaque contour. Puis un algorithme de diffusion basé *inpainting* est utilisé pour interpoler les données manquantes. Un codage encore sans perte des contours de la carte de profondeur est également proposé dans [67]. Il est basé sur un premier partitionnement des blocs contenant de discontinuités en sous blocs en forme arbitraire afin de définir un vecteur mouvement propre à chaque sous-bloc, suivi d'un codage arithmétique sans perte des valeurs des pixels délimitant le contour.

3.5.2 *Méthodes exploitant les corrélations entre profondeur et texture*

Nous avons déjà évoqué le fait qu'il existe une forte corrélation entre les deux composantes de texture et de profondeur, puisqu'elles représentent une projection de la même scène d'un même point de vue et à un même instant. Ainsi, afin de coder efficacement la profondeur, différents outils exploitant cette corrélation ont récemment été développés.

3.5.2.1 *Sélection des modes de prédiction*

Un premier type d'approches consiste à sélectionner les modes de prédiction des différents blocs de profondeur, en fonction des informations déjà décodées pour la texture. Dans cet esprit, l'outil *Depth Block Skip* fut présenté suite à un CfP de MPEG pour la vidéo 3D [68]. Le mode Skip est généralement utilisé à bas débits. Le principe est que lorsque l'erreur de prédiction entre le bloc courant et le bloc de référence co-localisé est jugée suffisamment faible (SSE inférieure à un seuil), la valeur du bloc co-localisé est simplement recopiée. Ce type de prédiction nécessite de transmettre le mode, sans autre information. L'extension à la composante de profondeur consiste simplement à utiliser le même mode Skip s'il est appliqué pour la texture. Dans ce cas, aucune donnée supplémentaire n'est transmise au décodeur (voir FIGURE 57).

Cet outil réduit ainsi également le nombre d'estimations de mouvements complexes à tester, et donc réduit de même la complexité de l'encodeur.

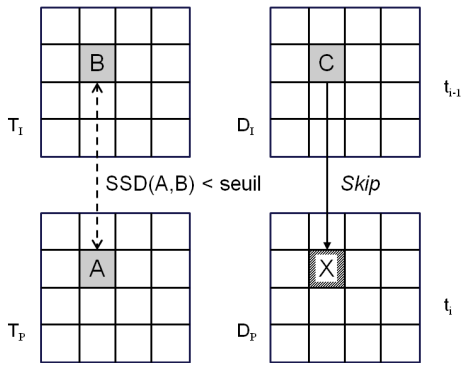


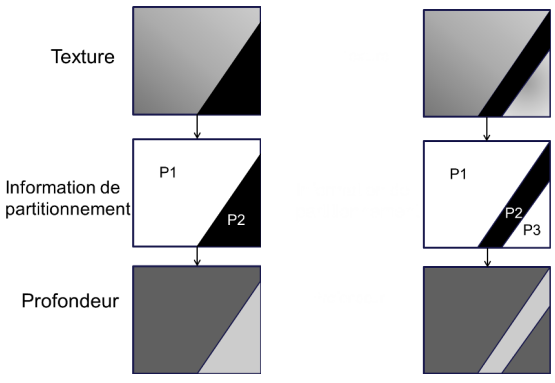
FIGURE 57. L'outil *Depth block Skip*.

3.5.2.2 *Héritage des informations de prédiction*

Grâce à la corrélation entre la texture et la profondeur, notamment aux alentours des contours, l'information de mouvement peut être partagée entre les deux composantes. *Seo et al.* proposent dans [69], un nouveau mode dénoté *Motion Sharing*. Dans ce nouveau mode, l'information de mouvement (vecteurs de mouvement + indices de trames de référence) d'un bloc de profondeur est directement héritée du bloc de texture correspondant. Le mode *Motion Sharing* n'est pas forcé sur le bloc de profondeur, mais il est comparé avec les différents modes existants (*Intra*, *Inter*). Le mode optimisant le coût débit-distorsion sera ensuite sélectionné.

Un outil similaire est développé pour le 3D-HEVC dans [70], dénoté *Motion Parameter Inheritance* MPI. Ici, il ne s'agit pas de créer un nouveau mode, mais simplement de considérer l'information de texture comme un nouveau candidat pour le mode MERGE (voir sous-section 3.2.3).

En mode Intra, l'information de partitionnement de la profondeur peut être aussi héritée de la texture. Dans le cas d'approximation du bloc de profondeur en deux ou trois régions constantes séparées par des lignes droites (*Depth Map Modeling DMM*), les modes 3 et 4 (deux autres nouveaux modes ajoutés à 3D-HEVC) permettent d'hériter directement l'information de partitionnement de la texture. Dans cette méthode, chaque échantillon du bloc de Luminance de la texture de référence est comparé à la moyenne du bloc lui même pour déterminer s'il appartient à la région P_1, P_2 ou P_3. Ce partitionnement sera utilisé tel quel pour le bloc de profondeur (voir FIGURE 58). Ainsi, ni l'information de partitionnement ni le seuil sont transmis au décodeur. En revanche, les constantes P_1, P_2 et P_3 doivent toujours être transmises.



(a) DMM mode 3 (b) DMM mode 4

FIGURE 58. Modes de modélisations de profondeur 3 et 4 dans 3D-HEVC.

Un outil similaire de partitionnement de la carte de profondeur en fonction de la texture est proposé dans [71]. La segmentation de la texture est utilisée pour diviser la carte de profondeur en un ensemble de régions. Ensuite, chaque région est supposée correspondre à une surface plane. Cette dernière est ensuite représentée par des coefficients codés par une méthode de codage standard telle que le H.264/AVC (voir FIGURE 59).

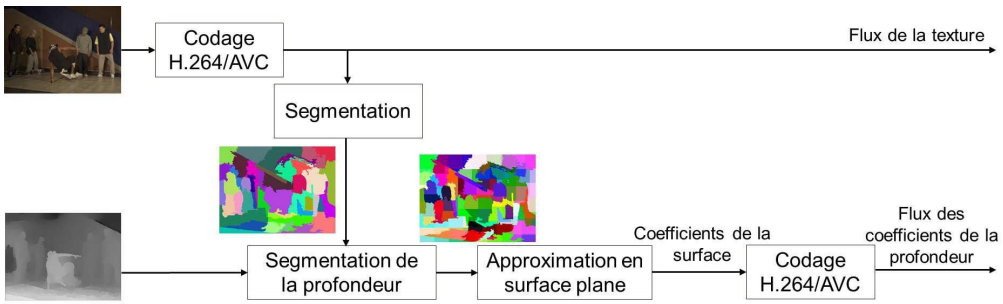


FIGURE 59. Schéma de codage de la profondeur exploitant la segmentation de la texture.

3.5.2.3 *Transformées spatiales*

Des transformées spatiales adaptées à la profondeur peuvent être reconstruites en fonction de la texture. Par exemple, *Daribo et al.* proposent dans [72] un modèle de *lifting* en ondelettes adaptatif. La commutation entre les filtres longs sur les contours et les filtres courts dans les zones homogènes de la carte de profondeur, est décidée en se basant sur les contours détectés dans l'image de texture.

3.5.3 *Méthodes optimisant le codage de la profondeur pour la qualité des vues synthétisées*

Puisqu'une carte de profondeur n'est pas affichée sur l'écran, mais utilisée plutôt pour synthétiser des vues virtuelles qui sont elles affichées, l'efficacité de compression du codeur de cartes de profondeur doit considérer la qualité de la vue synthétisée. Le principe se base sur le fait que les distorsions dans la carte de profondeur induisent des artéfacts dans la vue synthétisée, notamment au niveau des discontinuités et sur les bords des objets [73, 74, 75]. Aussi, l'optimisation débit-distorsion lors du codage des cartes de profondeur considèrent directement la distorsion de la vue intermédiaire synthétisée. Dans ce contexte, certains travaux estiment la distorsion de la vue synthétisée en fonction de la qualité de la carte de profondeur reconstruite. D'autres impliquent directement une phase de synthèse de vues lors du codage de la carte de profondeur. Nous discutons ces méthodes dans les sous-sections suivantes.

3.5.3.1 *Optimisation des synthèses de vues*

Le concept d'optimisation de la qualité de la vue synthétisée lors du codage de la carte de profondeur peut être utilisé en option dans le standard 3D-HEVC. L'outil utilisé est noté *Synthesized View Distortion Change* SVDC [57]. Le SVDC est défini par la différence de distorsion entre deux vues synthétisées à partir d'un bloc de profondeur codé différemment. La FIGURE 60 montre comment la métrique SVDC est calculée. Dans un premier temps, une vue v_{ref} est synthétisée à partir de la carte de profondeur originale d_{orig}. Ensuite, une vue v_1 est synthétisée à partir de la carte de profondeur d_1 où le bloc courant à coder et tous les blocs ultérieurs sont dans leur forme originale, tous les bloc antérieurs étant déjà reconstruits. Une autre vue v_2 est synthétisée à partir de la carte de profondeur d_2 semblable à la carte d_1 excepté que le bloc de profondeur courant est codé et reconstruit avec le mode à tester. Dans un second temps, la somme des erreurs quadratiques SSE est calculée entre la vue synthétisée v_{ref} et les vues v_1 et v_2, pour estimer les distorsions D_1 et D_2 des deux vues, respectivement. La différence entre les distorsions D_1 et D_2 est égale à la SVDC. Le mode optimisant le débit-SVDC sera choisi pour coder le bloc de profondeur courant.

Pour éviter une trop grande complexité apportée par l'opération de synthèse de vue, une méthode simplifiée est mise en œuvre, permettant simplement la synthèse des régions affectées par l'encodage du bloc courant (i.e. une synthèse partielle).

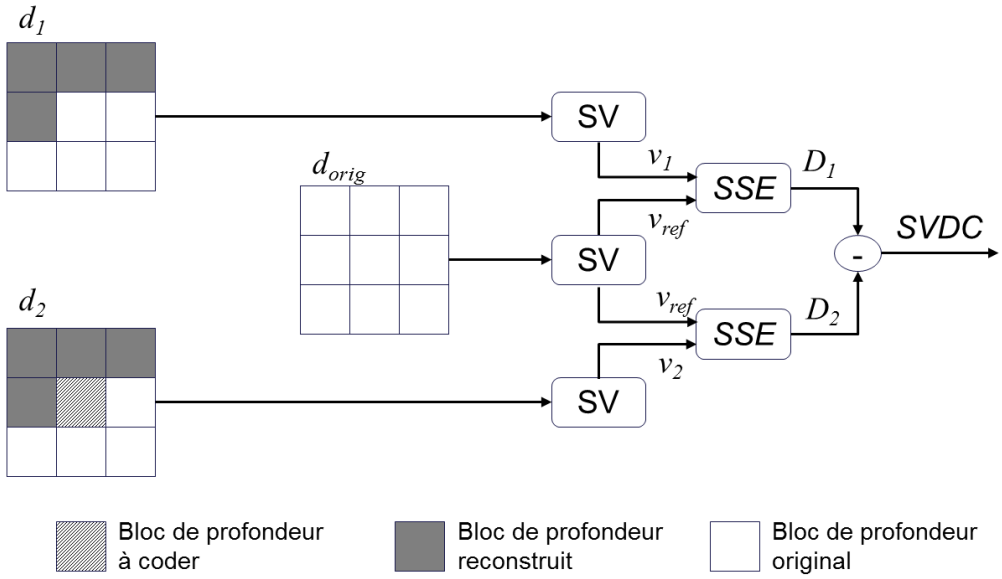


FIGURE 60. Calcul de SVDC.

Les avantages d'une telle méthode sont la précision dans la mesure de la distorsion qui prend en compte les occlusions et les disocclusions dans les vues synthétisées (e.g. la distorsion d'un certain bloc n'a pas d'effet sur la vue synthétisée puisque ce bloc est occulté dans la vue intermédiaire ou inversement). L'inconvénient principal de cette méthode est la complexité due aux opérations nécessaires pour la synthèse de vue, et qui sont répétées pour chaque bloc et pour chaque mode.

3.5.3.2 *Modèles de distorsion*

D'autres outils pour l'optimisation de la qualité de la vue synthétisée lors du codage de la profondeur, n'effectuent pas la synthèse de vue à proprement parler, mais viennent simplement estimer sa distorsion. La distorsion sur la vue synthétisée y est juste estimée. Ces outils lient ainsi la distorsion entre la profondeur et celle de la vue synthétisée. Plusieurs modèles existent dans la littérature. Des modèles de distorsion sur la vue synthétisée, que l'on nommera $\mathrm{Dist}_{\mathrm{Synth}}$, en fonction de la distorsion sur la profondeur $\mathrm{Dist}_{\mathrm{Prof}}$, ont été proposés dans [76]. Le modèle proposé dans [76] est basé sur le principe que la distorsion sur la profondeur entraîne une erreur de position dans la vue synthétisée :

$$\mathrm{Dist}^2_{\mathrm{Synth}} = \gamma \mathrm{Dist}_{\mathrm{Prof}} \tag{13}$$

où γ est fonction des paramètres de la caméra.

L'avantage principal de ces modèles d'estimation de distorsion est bien sûr qu'ils sont moins complexes que les solutions qui synthétisent effectivement les vues lors de l'encodage de la carte de profondeur. En contrepartie, leurs performances sont moins bonnes.

3.6 CONCLUSION

Ce chapitre a introduit plusieurs outils de codage de la carte de profondeur proposés dans la littérature, répartis suivant trois catégories selon *Lucas et al.* [20]. Certaines approches se basent sur les caractéristiques intrinsèques d'une carte de profondeur telles que sa représentation en régions lisses séparées par des contours. D'autres outils exploitent la corrélation de la carte de profondeur avec la texture associée. Le choix du mode de prédiction peut ainsi être pris en fonction de l'information de texture, des informations de prédiction peuvent être héritées

de la texture, ou encore des transformées spatiales peuvent être conçues spécifiquement pour la profondeur en se basant sur des informations de la texture. Enfin, certaines approches proposent d'optimiser le codage de la carte de profondeur en fonction de la qualité de ce qui va être vraiment affiché, c.à.d la vue synthétisée. Des modèles de distorsions sont construits afin de lier la distorsion de la profondeur avec la distorsion sur la vue synthétisée pour qu'elle puisse être estimée lors de l'encodage de la carte de profondeur.

Le chapitre suivant présente notre contribution dans la compression de la carte de profondeur. Nous proposons un outil de compression de la carte de profondeur qui tient profit des avantages des trois catégories de codage présentées dans ce chapitre. Avec une simple implémentation, l'outil de compression de profondeur proposé, respecte les caractéristiques intrinsèques de la carte de profondeur et exploite la corrélation avec la texture pour l'encodage de la carte de profondeur.

CONTRIBUTIONS

4 MÉTHODE DE CODAGE 3D JOINT TEXTURE/PROFONDEUR

La visualisation d'un même organisme
dans sa totalité en trois dimensions
devrait aussi apporter une meilleure
appréhension de divers phénomènes
en embryologie ou en biologie cellulaire.

— Aassif Benassarou *et al.*, Visualisation 3D relief du vivant [20]

Objectifs spécifiques du chapitre :

- **Connaître** et **Comprendre** le codeur 2D adopté.
- **Synthétiser** un schéma de codage **joint** 2D+Z **scalable** basé sur le codeur 2D.
- **Évaluer** le schéma proposé.

4.1 INTRODUCTION

Comme mentionné dans le chapitre précédent, ce chapitre présente notre contribution en termes de compression de la carte de profondeur. Nous proposons ainsi un outil adapté qui tire profit des avantages des trois catégories de codage présentées dans le chapitre précédent. L'outil de compression de profondeur proposé respecte les caractéristiques intrinsèques de la carte de profondeur, et exploite la corrélation avec la texture pour l'encodage de la carte de profondeur, tout en optimisant la qualité de la vue synthétisée.

Ainsi, afin de respecter les caractéristiques intrinsèques de la carte de profondeur, nous avons proposé une extension de la méthode de compression LAR (*Locally Adaptive Resolution*) [77]. Le LAR est un codec d'images 2D développé par l'équipe Image de l'IETR. Il est basé sur l'idée que la résolution locale d'une image dépend de son activité locale. Ce codec préserve en particulier les contours des objets même à bas débits, ce qui garantit une bonne qualité visuelle. Le LAR constitue ainsi un bon candidat pour la compression des cartes de profondeur.

L'extension proposée du codec LAR consiste en un schéma scalable de compression 2D+Z. Dans un premier temps, la profondeur et la texture sont codées conjointement à basse résolution. Ensuite, la qualité de la texture est raffinée.

Ce chapitre présente donc dans un premier temps, la plateforme du LAR 2D (Section 4.2). Ensuite, nous présentons en global le schéma scalable de compression 2D+Z (Section 4.3). Ensuite, dans la Section (4.4), nous détaillons le schéma de codage à basse résolution, tout en focalisant sur l'outil proposé pour la compression de la carte de profondeur. La Section 4.5 présente ensuite les résultats d'évaluation du schéma de codage à basse résolution. La Section 4.6 détaille enfin le schéma de raffinement et ses résultats d'expérimentation.

4.2 PLATEFROME LAR 2D

Le LAR 2D a été développée à l'IETR. c'est la base de l'algorithme de codage scalable proposé. Dans cette section nous en faisons une présentation simplifiée. Pour plus de détails, voir [78] et [77]. Le schéma global du LAR repose sur un partitionnement en blocs carrés de taille variable (*QuadTree*), suivi d'une décomposition pyramidale avec transformation et prédiction entrelacées (voir FIGURE 61). Plusieurs fonctionnalités avancées ont été ajoutées au LAR telles que la robustesse aux erreurs, le codage avec ou sans pertes, ou l'optimisation débit-distorsion [79]. Dans les sous-sections suivantes, nous détaillons les différentes étapes du LAR.

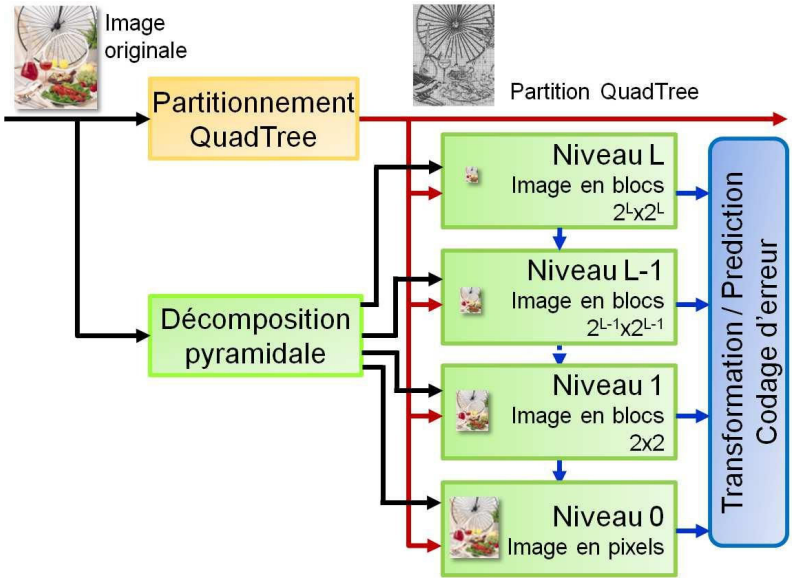
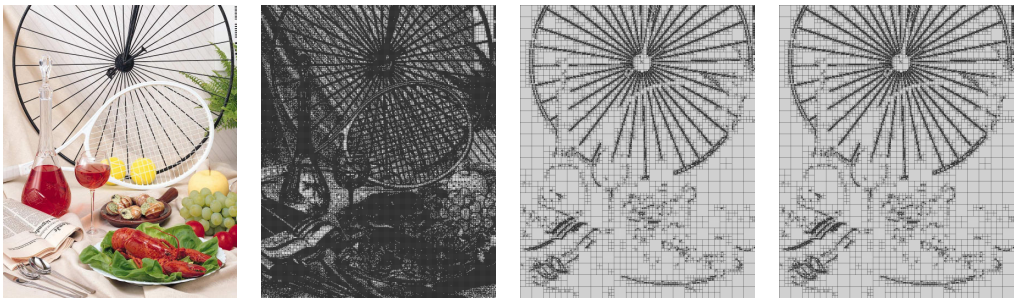


FIGURE 61. Schéma multi-résolution du LAR.

4.2.1 *Partitionnement QuadTree*

Le LAR débute par une analyse de l'activité locale de l'image originale I de taille $N_x \times N_y$, menant à une partition ou une grille en blocs de taille variable (1×1, 2×2, ..., 64×64, 128×128 pixels). L'estimation de la taille du bloc est effectuée via un critère d'homogénéité, en comparant le gradient du bloc (la différence entre la valeur maximale locale et la valeur minimale locale du bloc) à un seuil donné Th_{Quad}. Le bloc dont le gradient est supérieur au seuil Th_{Quad}, sera divisé en 4 sous-blocs. La taille du bloc définit ainsi la résolution locale. Celle ci dépend donc du critère d'homogénéité : les petits blocs se situent sur les contours, alors que les grands blocs se localisent dans les zones homogènes (voir FIGURE 62). Plus Th_{Quad} est élevé, plus la grille va contenir de grands blocs (degré de granularité faible) et inversement (plus Th_{Quad} est petit, plus la grille va contenir de petits blocs) et le degré de granularité est plus élevé.



(a) Image originale (b) $Th_{Quad} = 15$ (c) $Th_{Quad} = 55$ (d) $Th_{Quad} = 170$

FIGURE 62. Exemple de partitionnement *QuadTree* de l'image naturelle "bike" pour différents seuils Th_{Quad}.

4.2.2 *Décomposition pyramidale*

Après le partitionnement *QuadTree*, le LAR utilise une représentation pyramidale multi-résolution de l'image : l'image originale est décomposée d'une manière dyadique pour avoir L niveaux de résolution, où le niveau $l = 0$ de la pyramide est le niveau de pleine résolution où la taille du bloc est de 1x1 pixel, et le niveau supérieur L de la pyramide est le niveau de plus basse résolution (voir FIGURE 61). La décomposition dyadique consiste à sous-échantillonner l'image

pour passer du niveau l au niveau $l+1$: un bloc 2x2 à un niveau l de la pyramide est remplacé au niveau $l+1$, par un seul pixel dont la valeur est la valeur moyenne des deux pixels de la première diagonale.

4.2.3 *Transformation et Prédiction*

Après la décomposition dyadique de l'image, une phase de transformation et de prédiction est conditionnée par le *QuadTree* (voir Fig 63). À un niveau l :

- si la résolution locale d'un bloc (définie par sa taille dans le *QuadTree*) est inférieure à la résolution du niveau courant l (i.e si la taille du bloc, déterminée par le QuadTree, est $> 2^l$), la valeur du bloc au niveau $l+1$ est simplement recopié dans les 4 sous-blocs du niveau l.
- si la résolution locale du bloc est supérieure ou égale à la résolution du niveau courant (i.e. si la taille du bloc, déterminée par le QuadTree, est $\leqslant 2^l$), le bloc sera découpé en 4 sous-blocs, dont les valeurs doivent être prédites.

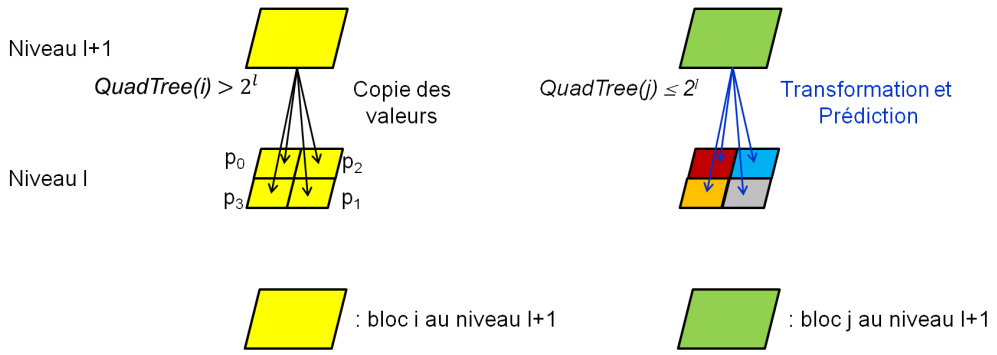


FIGURE 63. Phase de transformation et de prédiction conditionnée par le *QuadTree*.

Ensuite, une transformation S+P (Transformation en S + Prédiction) [80], est appliquée sur les sous-blocs à prédire, à chaque niveau de la pyramide. La transformation en S prend comme entrée les deux vecteurs formés par les sous-blocs diagonalement adjacents d'un bloc décomposé en 4 sous-blocs et fait sortir 4 coefficients(voir FIGURE 64). Si (p_0,p_1) sont les valeurs de sous-blocs de la première diagonale, et (p_2,p_3) ceux de la deuxième diagonale, alors les coefficients de la transformation en S (u_0,u_1, u_2 et u_3) sont calculés suivant l'équation (14), où (u_0, u_1) et (u_2, u_3) représentent la moyenne et le gradient de la première et la deuxième diagonale du bloc décomposé, respectivement.

$$\begin{aligned} u_0 &= \lfloor (p_0 + p_1)/2 \rfloor. & u_2 &= \lfloor (p_2 + p_3)/2 \rfloor. \\ u_1 &= p_1 - p_0. & u_3 &= p_3 - p_2. \end{aligned} \tag{14}$$

En utilisant la transformation S+P à chaque niveau de la pyramide, les coefficients u_0 d'un certain niveau sont automatiquement hérités du niveau supérieur. Ainsi, seuls les trois coefficients u_1, u_2 et u_3 doivent être estimés à chaque niveau. Ainsi, la transformation en S est suivie par une étape de prédiction entre les niveaux de la pyramide, de deux valeurs gradients (u_1, u_3) et d'une seule valeur moyenne (u_2). La prédiction utilisée est celle de Wu [80]. Il s'agit d'une prédiction en deux passes, utilisant une combinaison linéaire des blocs voisins reconstruits et des blocs du niveau le plus haut. Les erreurs de prédiction sont ensuite quantifiées, en utilisant un paramètre d'entrée de quantification Qp. Le facteur de quantification de chaque niveau de la pyramide est donné comme suit : $Q_l = \frac{Q_p}{2^l}$, avec l représente le niveau courant de la pyramide ($l = 0$ représente le niveau de pleine résolution). Enfin, l'erreur de prédiction quantifiée est encodée avec un codeur entropique.

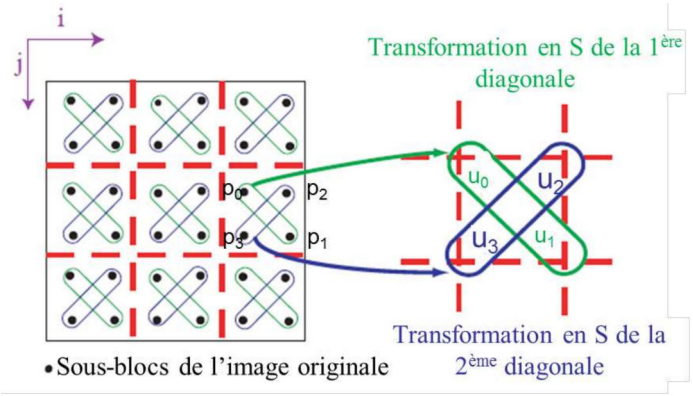


FIGURE 64. Transformation en S.

4.2.4 *Post-traitement*

Afin d'éliminer les effets de bloc, un post-traitement est appliqué à l'image reconstruite. Ce post-traitement consiste en une interpolation linéaire classique, *Bi-linear filter*, suivant l'algorithme 4.2.1. Il permet de lisser les zones homogènes tout en conservant les contours, suivant un certain seuil Th_{PT}, choisi égal à Th_{Quad}. Pour un pixel X, si la différence entre le pixel courant et son voisin est supérieure à Th_{PT}, ce voisin n'est pas pris en compte dans l'interpolation. Au contraire, si la différence est inférieure au Th_{PT}, il sera pris en compte dans l'interpolation. Enfin, la valeur du pixel courant est remplacée par la moyenne des pixels voisins pris en compte.

Algorithme 4.2.1 : ***Interpolation classique***

Entrée : X, A, B, C, D
Sortie : X'
$w_A = w_B = w_C = w_D = 1$
si $(|X - A| \geqslant Th_{PT})$ $w_A = 0$;
si $(|X - B| \geqslant Th_{PT})$ $w_B = 0$;
si $(|X - C| \geqslant Th_{PT})$ $w_C = 0$;
si $(|X - D| \geqslant Th_{PT})$ $w_D = 0$;
$X' = Moyenne(w_A * A,\ w_B * B,\ w_C * C,\ w_D * D)$;
{retourner la nouvelle valeur X'}
retourner X'

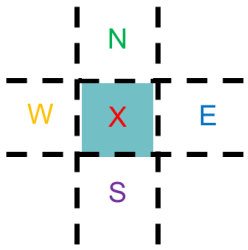


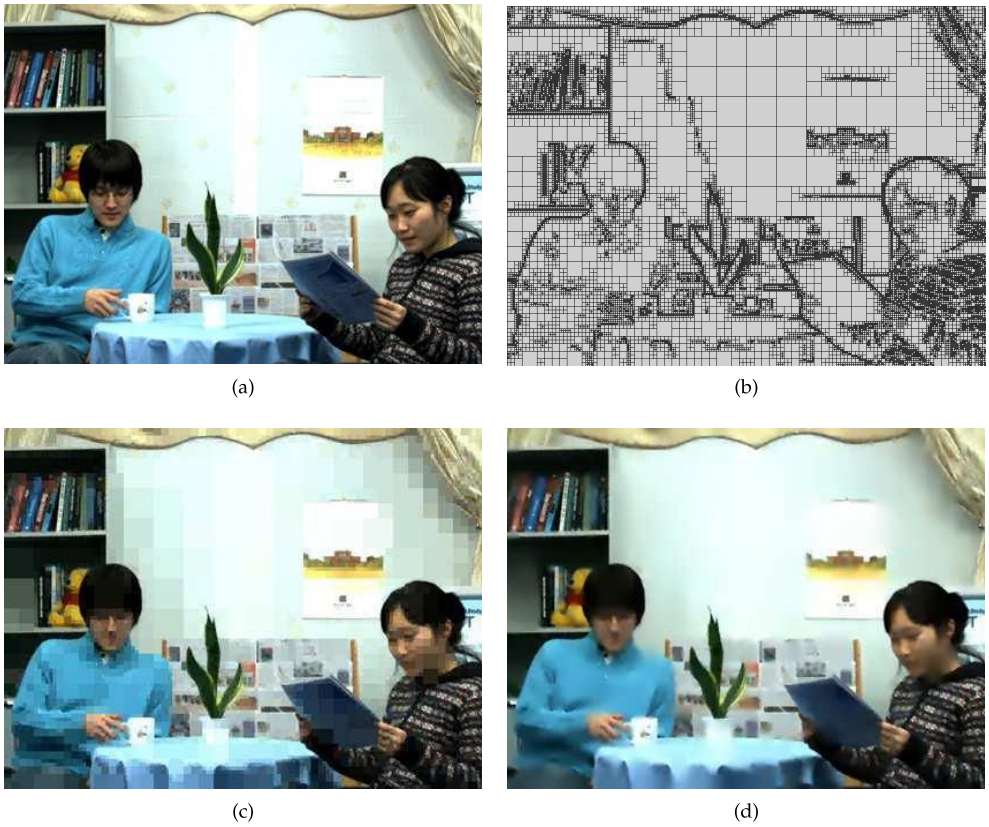
FIGURE 65. Pixel courant X et les 4 voisins connexes N, E, S et W.

Le LAR est un codeur développé initialement pour la compression des images naturelles 2D. En raison de ses caractéristiques, le LAR peut être potentiellement utilisé pour la compression des cartes de profondeur. Une extension directe du LAR 2D aux données 2D+Z, consiste à coder la texture et la profondeur indépendamment. Pour une compression plus efficace, nous proposons un schéma de codage scalable joint texture/profondeur.

4.3 SCHÉMA GLOBAL DE CODAGE SCALABLE ET JOINT TEXTURE/PROFONDEUR

4.3.1 *Principe du schéma proposé*

Comme mentionné au début de ce mémoire, la multitude des disciplines utilisant la technologie 3D entraîne une hétérogénéité dans la qualité des images 3D. Ceci nécessite un schéma de codage 3D scalable capable d'adapter le flux de données au canal de transmission. Outre la scalabilité, la compression des données 2D+Z nécessite un schéma de codage joint texture/profondeur afin



(a) (b)

(c) (d)

FIGURE 66. Exemple de codage de Newspaper view 6 frame 1 avec le LAR à 0.25 bpp : (a) image originale ; (b) grille de partitionnement avec $\mathrm{Th}_{\mathrm{Quad}} = 38$, $\mathrm{Qp} = 25$; (c) image reconstruite avec effets de blocs (PSNR = 28.99dB) ; (d) image reconstruite avec post-traitement, $\mathrm{Th}_{\mathrm{PT}} = 38$, (PSNR = 30.06dB).

de préserver la cohérence entre ces deux composantes et d'assurer ainsi une haute qualité de synthèse et d'affichage. Nous proposons ainsi un schéma de codage 2D+Z **scalable** et **joint** texture/profondeur basé sur le LAR. Ce schéma proposé se constitue de deux étapes : 1) un codage à basse résolution de la profondeur et de la texture qui va fournir la profondeur et la texture à basse résolution (Section 4.4), et 2) un codage à haute résolution de la texture qui va ensuite raffiner la texture (Section 4.6) (voir FIGURE 67).

Or, la résolution de l'image de sortie du LAR, est contrôlée par le degré de granularité de la grille fournie par le partitionnement QuadTree. Ainsi, le choix de la grille pour le codage des données 2D+Z, constitue la clé dans le schéma de codage scalable joint texture/profondeur. Plusieurs choix sont possibles pour la grille de partitionnement. Elle peut être construite soit à partir de la profondeur uniquement, soit à partir de la texture uniquement, soit à partir de la texture et profondeur conjointement.

1 codage à basse résolution : en comparaison avec la texture, la carte de profondeur contenant peu de contenu et plus de contours, la grille de profondeur contient plus de grands blocs mais garde bien les contours. Ainsi, le schéma de codage à basse résolution prend comme entrée la grille de profondeur avec la profondeur et la texture originales. Il fournit ainsi la profondeur et la texture à basse résolution. Dans cette étape, un outil de compression joint texture/profondeur, appelé LARP (LAR pour Profondeur) est proposé. Le LARP exploite la corrélation entre la texture et la profondeur pour coder efficacement la profondeur.

2 codage à haute résolution : le schéma de codage à haute résolution prend comme entrée la texture originale, la texture à basse résolution et la grille profondeur plus texture et il fournit un flux de raffinement pour l'image de texture. Il permet ainsi de faire un rehaussement de la qualité de la texture.

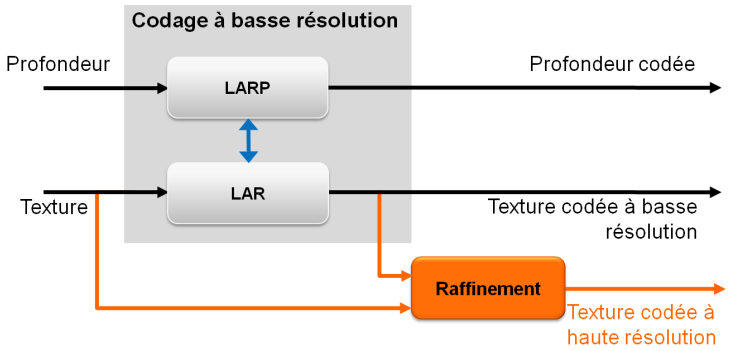


FIGURE 67. Schéma global de codage scalable 2D+Z proposé.

4.3.2 *Choix des paramètres du LAR*

Étant basé sur le LAR, les paramètres d'initialisation à contrôler sont le seuil de partitionnement *QuadTree* $\mathrm{Th}_{\mathrm{Quad}}$ et le facteur de quantification $\mathrm{Q_p}$. La problématique du choix des paramètres $\mathrm{Th}_{\mathrm{Quad}}$ et $\mathrm{Q_p}$ a été étudiée pour les images naturelles 2D dans [81]. Un modèle à basse complexité a été proposé pour choisir ces paramètres. Pour le codage des données 2D+Z, des expérimentations dans [79] ont permis de déduire que la relation optimale liant le couple $\{\mathrm{Th}_{\mathrm{Quad}};\ \mathrm{Q_p}\}$ est une fonction linéaire.

Dans les sections suivantes (Section 4.4, Section 4.5, et Section 4.6), nous détaillons respectivement les deux étapes du schéma de codage 2D+Z scalable et joint texture/profondeur.

4.4 CODAGE À BASSE RÉSOLUTION

4.4.1 *Principe*

La première étape du schéma de codage scalable proposé, est la compression à basse résolution de la texture, et la compression de la profondeur (voir FIGURE 68).

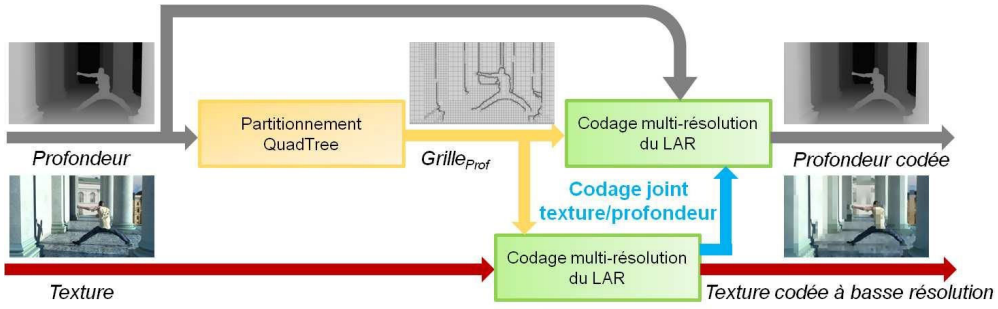


FIGURE 68. Première étape du codage scalable proposé : codage à basse résolution de la texture et codage de la profondeur avec un outil de compression joint texture/profondeur.

Dans un premier temps, le partitionnement *QuadTree* est réalisé à partir de la carte de profondeur, avec un seuil donné $\mathrm{Th}_{\mathrm{Quad}}$. La grille de partition résultante $\mathrm{Grille}_{\mathrm{Prof}}$ est formée de grands blocs dans les zones homogènes de la carte de profondeur, mais préserve les contours des objets qui sont représentés par des petits blocs.

Ensuite, en se basant sur la grille de profondeur $\mathrm{Grille}_{\mathrm{Prof}}$, la texture est codée à bas débit, avec le schéma de multi-résolution du LAR (voir Section 4.2). Nous obtenons ainsi une représentation à basse résolution basée bloc de la texture, mais qui respecte les formes des objets présents dans la carte de profondeur.

Enfin, plusieurs pistes sont envisageables pour coder efficacement la profondeur. Nous proposons d'exploiter la forte corrélation entre la profondeur et la texture. Cette proposition est inspirée de la corrélation qui existe entre les différentes composantes couleurs d'une image naturelle [82]. En effet, dans un schéma de codage prédictif, une relation linéaire existe entre les

erreurs de prédiction des différentes composantes couleurs. Une décorrélation adaptative a ainsi été développée dans [83] afin de coder efficacement les images naturelles 2D. Elle consiste à prédire l'erreur de prédiction d'une composante couleur à partir de celle d'une autre composante couleur, permettant ainsi de réduire le débit généré. L'efficacité de cette technique sur le codage des images couleurs nous a encouragés à faire une extension de cette technique à la composante de profondeur. Nous avons proposé ainsi dans un premier temps d'étendre cette décorrélation à la profondeur. Néanmoins, elle ne donne pas de bons résultats. Les erreurs de prédiction de la profondeur ne sont pas parfaitement corrélées aux erreurs de prédiction des composantes couleurs.

Ensuite, nous avons proposé une autre approche de codage de profondeur, appelée LARP, comportant deux techniques : une technique de codage de la profondeur exploitant sa corrélation avec la texture, appelée "Meilleur Prédicteur", suivie par un post-traitement adaptatif sur la carte de profondeur reconstruite. Une telle approche donne de bons résultats objectifs ainsi que visuels. Dans les sous-sections suivantes, nous détaillons la technique du Meilleur Prédicteur (Section 4.4.2) et le post-traitement adaptatif (Section 4.4.3).

4.4.2 *Meilleur Prédicteur*

La technique "Meilleur Prédicteur" consiste à améliorer la prédiction de la composante de profondeur Z, en utilisant le meilleur prédicteur de la composante de luminance Y déjà codée de l'image texture associée. Nous rappelons que la composante de luminance Y est déjà prédite par un prédicteur par défaut qui est le prédicteur de Wu. On introduit les notations suivantes.

- Z_i : valeur du gradient de la profondeur à un bloc i à prédire dans la carte de profondeur. Z_i peut représenter u_1 ou u_3 comme montré dans la FIGURE 64.
- Y_i : valeur du gradient de la luminance au bloc i dans l'image de texture associée.
- $\widetilde{Y_i}^j$: valeur prédite de Y_i par le prédicteur j,
  $j \in [0, ..., \text{NbPrédicteurs}]$ avec j=0 est le prédicteur par défaut.
- $\widehat{Y_i}^j$: valeur reconstruite de $\widetilde{Y_i}^j$.
- $\texttt{prediction}(Y_i, j)$: processus retournant $\widetilde{Y_i}^j$.
- $E_i^j = |\widehat{Y_i}^j - \widetilde{Y_i}^j|$: erreur de prédiction par le prédicteur j.

Alors que la FIGURE (69a) illustre un schéma de codage de type simulcast des données 2D+Z à un certain niveau l de la pyramide , où la profondeur Z et la luminance Y de la texture associée, sont codées indépendamment, la FIGURE (69b) illustre l'approche de "Meilleur Prédicteur" pour la compression jointe texture/profondeur.

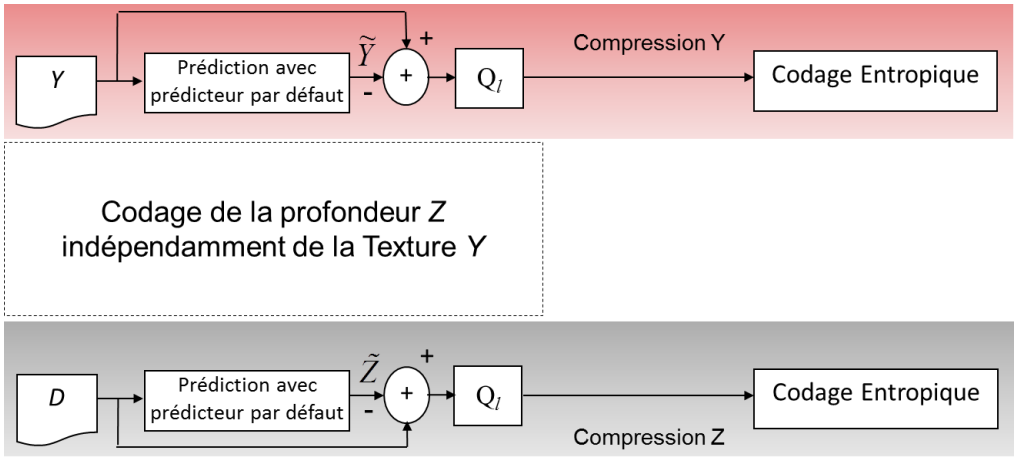
Cette technique repose sur trois étapes.

1) Dans un premier temps, après la reconstruction de la valeur $\widehat{Y_i^0}$ (qui est la même valeur reconstruite au décodeur), on prédit à nouveau la valeur de $\widehat{Y_i}$ (prédiction a posteriori) avec un ensemble déjà défini de prédicteurs. Pour réduire la complexité de calcul, le nombre de prédicteurs est limité au nombre de directions principales (voir FIGURE 70).

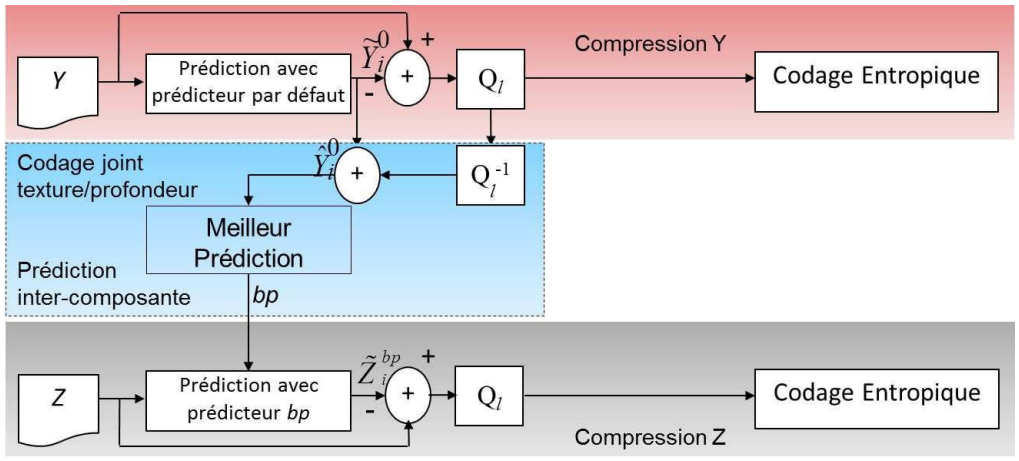
   Dans le cas d'une prédiction du gradient selon la première diagonale (u_1), les différents prédicteurs a posteriori sont les suivants (voir FIGURE 71) :

   - cas direction 1 (possibilité d'un contour horizontal) :
     $\texttt{prediction}(Y_i, 1) = \widetilde{Y_i}^1 = WW - SW$,
   - cas direction 3 (possibilité d'un contour vertical) :
     $\texttt{prediction}(Y_i, 3) = \widetilde{Y_i}^3 = NW - NE$,
   - cas direction 4 (possibilité d'un contour suivant la première diagonale) :
     $\texttt{prediction}(Y_i, 4) = \widetilde{Y_i}^4 = 0$.

   Dans le cas d'une prédiction du gradient selon la deuxième diagonale (u_3), les différents prédicteurs a posteriori sont les suivants :



(a) Codage 2D+Z indépendant.



(b) Codage de la profondeur avec la technique de "Meilleur Prédicteur".

FIGURE 69. Schéma de compression simulcast vs schéma de compression avec la technique de "Meilleur Prédicteur", à un niveau l de la pyramide de multi-résolution.

- cas direction 1 (possibilité d'un contour horizontal) :
  $\text{prediction}(Y_i, 1) = \widetilde{Y_i}^1 = W - S$,
- cas direction 2 (possibilité d'un contour suivant la deuxième diagonale) :
  $\text{prediction}(Y_i, 2) = \widetilde{Y_i}^2 = 0$,
- cas direction 3 (possibilité d'un contour vertical) :
  $\text{prediction}(Y_i, 3) = \widetilde{Y_i}^3 = S - W$.

2) Sélection du meilleur prédicteur : parmi ces différents prédicteurs, y compris le prédicteur par défaut (prédicteur de Wu), le prédicteur qui minimise la distance $|\widehat{Y_i}^j - \widetilde{Y_i}^0|$ est choisi comme meilleur prédicteur *bp* du bloc i.

3) Prédiction du pixel de la profondeur : ce meilleur prédicteur *bp* est utilisé sur le bloc i de la carte de profondeur associée, pour obtenir une meilleure prédiction de Z_i.

La FIGURE (69b) illustre l'implémentation de cette approche à un niveau l de la pyramide. L'algorithme 4.4.1 et l'organigramme de la FIGURE (72) spécifient la procédure de sélection du meilleur prédicteur.

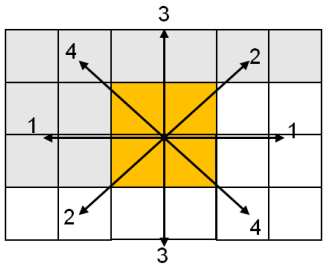
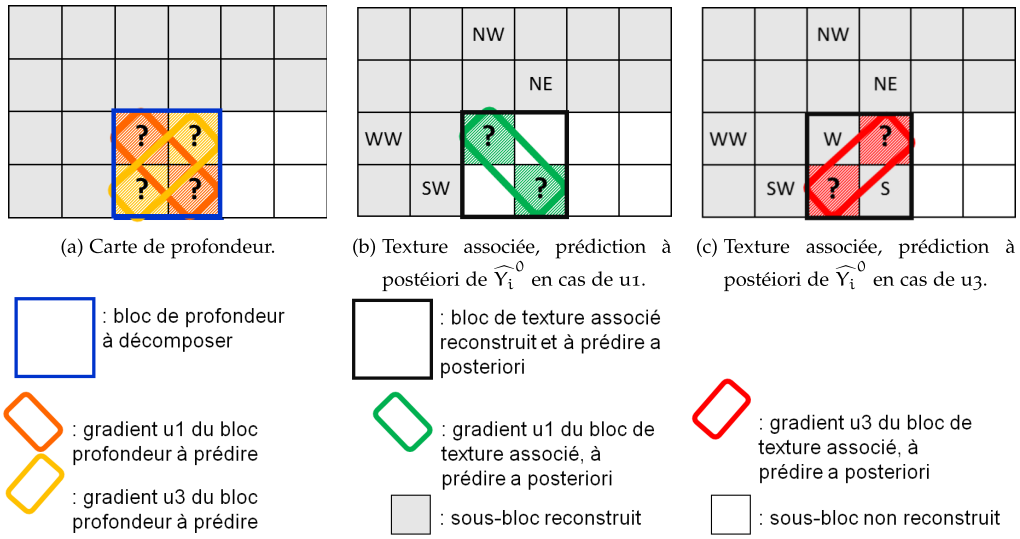


FIGURE 70. 4 directions possibles de prédiction a posteriori de $\widehat{Y_i}^0$.



(a) Carte de profondeur.

(b) Texture associée, prédiction à postéiori de $\widehat{Y_i}^0$ en cas de u1.

(c) Texture associée, prédiction à postéiori de $\widehat{Y_i}^0$ en cas de u3.

FIGURE 71. Prédiction a posteriori des gradients du bloc de texture associé au bloc i de profondeur à décomposer.

Algorithme 4.4.1 : *Meilleure Prédiction*

Entrée : Y_i ;$\widehat{Y_i}^0$
Sortie : bp
  $best = +\inf, bp = 0, j = 1$
  pour tout $j < Nb_{Prédicteurs}$ **faire**
    $\widetilde{Y_i}^j = \text{prédiction}(Y_i, j)$
    si $|\widetilde{Y_i}^j - \widehat{Y_i}^0| < best$ **alors**
    $best = |\widetilde{Y_i}^j - \widehat{Y_i}^0|$
    $bp = j$
    fin si
  fin pour
{retourner le meilleur prédicteur selon Y}
retourner bp

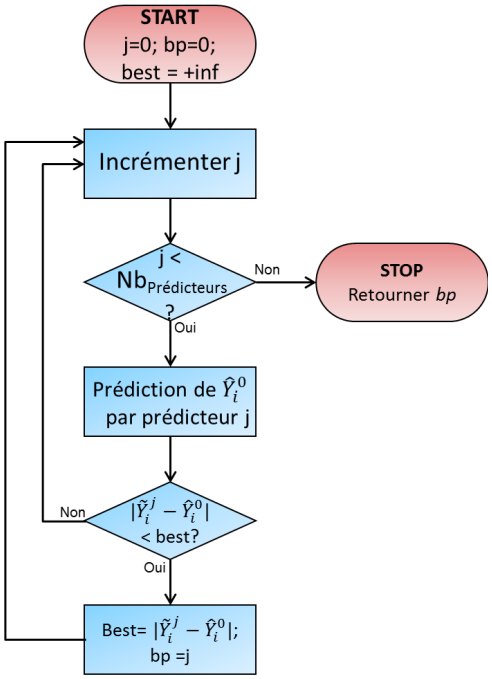
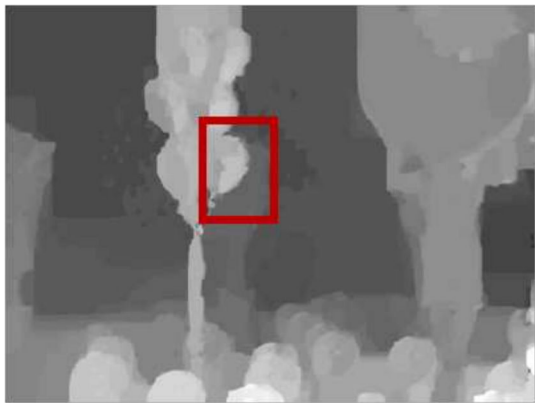


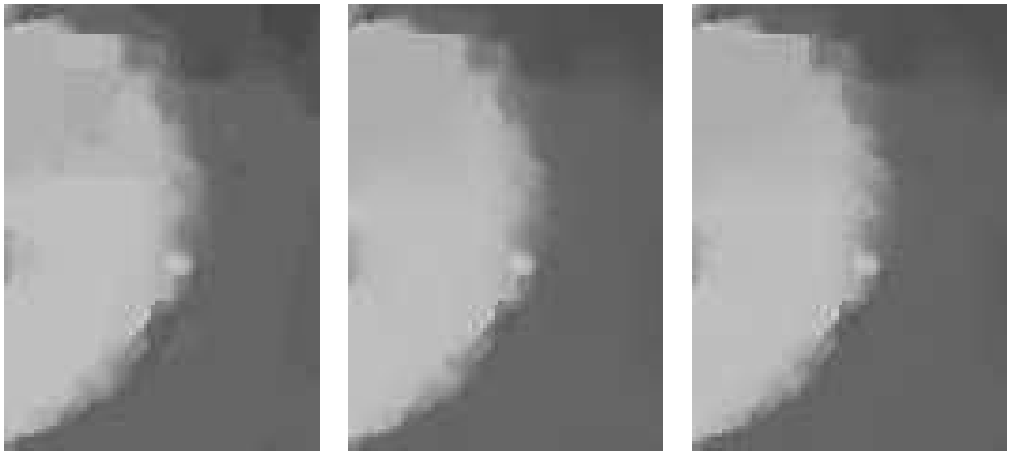
FIGURE 72. Procédure de sélection du meilleur prédicteur de $\widehat{Y_i}^0$.

Les FIGURES 73 et 74 illustrent quelques exemples de codage des cartes de profondeur avec le LAR sans et avec la technique de "Meilleur Prédicteur". La technique de "Meilleur Prédicteur" possède plusieurs avantages :

- Opérant sur les valeurs reconstruites, elle peut être dupliquée au décodeur, sans transmettre aucune information additionnelle.
- Elle est directement applicable dès lors que l'image de texture et l'image de profondeur partagent la même grille de partitionnement. C'est celle de la profondeur. D'une part, malgré qu'elle fait paraître des grands blocs dans les zones homogènes de profondeur, cette grille montre nettement, avec des petits blocs, les forts gradients séparant les différents plans dans la carte de profondeur. Une telle partition permet de conserver les contours lors du codage à basse résolution de la texture. Ceci permet ainsi une meilleure qualité de reconstruction. D'autre part, grâce à la corrélation entre la texture et la profondeur, notamment sur les contours, l'utilisation de la même grille, augmente l'efficacité de la technique de "Meilleur Prédicteur". Elle permet de sélectionner le prédicteur adéquat, diminuant par ceci l'erreur de prédiction et fournissant ainsi un gain en débit.



(a)

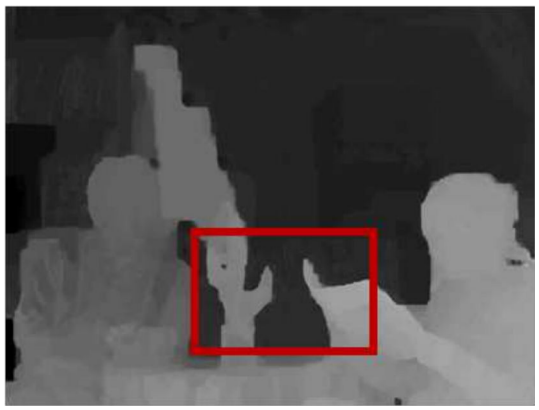


(b) (c) (d)

FIGURE 73. Carte de profondeur de Balloons vue 5 image 1 à 0.06 bpp avec $\{Q_p = 30;\ Th_{Quad} = 20\}$ (a) et (b) originale ; (c) codée avec le LAR classique (PSNR = 39 dB) ; (d) codée avec la technique de "Meilleur Prédicteur" (PSNR = 40 dB).

4.4.3 *Interpolation Adaptative*

Dans cette sous-section, nous détaillons la deuxième étape du LARP. Après reconstruction, la carte de profondeur est soumise à un post-traitement pour éliminer les effets blocs. L'interpolation classique du LAR est remplacée par une interpolation adaptative, plus adaptée aux



(a)

(b)

(c)

(d)

FIGURE 74. Carte de profondeur de Newspaper vue 6 image 1 à 0.06 bpp (a) et (b) originale ; (c) codée avec le LAR classique (PSNR = 35.5 dB) avec $\{Q_p = 64;\ Th_{Quad} = 42\}$; (d) codée avec la technique de "Meilleur Prédicteur" (PSNR = 36.1 dB) avec $\{Q_p = 71;\ Th_{Quad} = 47\}$.

caractéristiques des cartes de profondeur. En effet, comme celles-ci contiennent des gradients plus forts qu'une image de texture, il est important de les préserver pour la qualité des images reconstruites. L'idée est donc de lisser les zones homogènes, tout en gardant un gradient fort sur les contours. Nous introduisons ainsi une pondération sur les valeurs des pixels voisins (A, B, C et D) au pixel courant. L'algorithme d'interpolation linéaire adaptative est le suivant :

Algorithme 4.4.2 : ***Interpolation linéaire adaptative***

Entrée : $X,\ A,\ B,\ C,\ D$
Sortie : X'
w_A =**calcul_poids** (X, A)
w_B =**calcul_poids** (X, B)
w_C =**calcul_poids** (X, C)
w_D =**calcul_poids** (X, D)
$X' = Moyenne(w_A * A,\ w_B * B,\ w_C * C,\ w_D * D);$
{retourner la nouvelle valeur X'}
retourner X'

calcul_poids (X, V) **:**

Entrée : $X,\ V$

Sortie : w

$dif = |X - V|$

si $(dif \geqslant Th_{PT})\ w = 0;$

si $(\frac{4}{5}Th_{PT} \leqslant dif < Th_{PT})\ w = 0.2;$

si $(\frac{3}{5}Th_{PT} \leqslant dif < \frac{4}{5}Th_{PT})\ w = 0.4;$

si $(\frac{2}{5}Th_{PT} \leqslant dif < \frac{3}{5}Th_{PT})\ w = 0.6;$

si $(\frac{1}{5}Th_{PT} \leqslant dif < \frac{2}{5}Th_{PT})\ w = 0.8;$

si $(dif < \frac{1}{5}Th_{PT})\ w = 1;$

retourner w

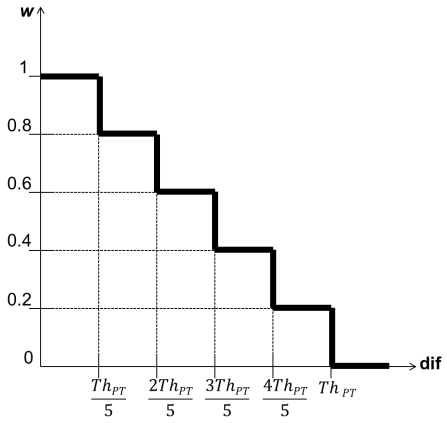


FIGURE 75. Calcul du poids du pixel voisin en fonction de la différence entre ce dernier et le pixel courant.

La différence entre la valeur du pixel courant X et la valeur du pixel voisin (A, B, C ou D) est comparée au seuil d'interpolation Th_{PT}. Le coefficient de pondération w pour chaque pixel voisin est ainsi donné par la FIGURE 75. La sélection du coefficient de pondération w est fonction de la différence entre la valeur du pixel courant et du pixel voisin dans la carte de profondeur. Plus le pixel voisin a une valeur proche du pixel courant dans la carte de profondeur, (i.e. plus les pixels appartiennent à des plans de profondeurs proches), plus il doit être l'effet d'interpolation. Inversement, plus les valeurs sont éloignées, signifiant que les pixels appartiennent à des plans de profondeurs différents, moins l'effet d'interpolation avec le pixel courant doit être. Ainsi, une distribution linéaire des coefficients en fonction de la différence entre les pixels est adoptée suivant la FIGURE 75. D'autre part, l'optimal est de choisir le seuil d'interpolation Th_{PT} une fonction linéaire de Th_{Quad}.

Ensuite, la valeur du pixel courant X est remplacée par la moyenne pondérée des pixels voisins selon l'équation 15.

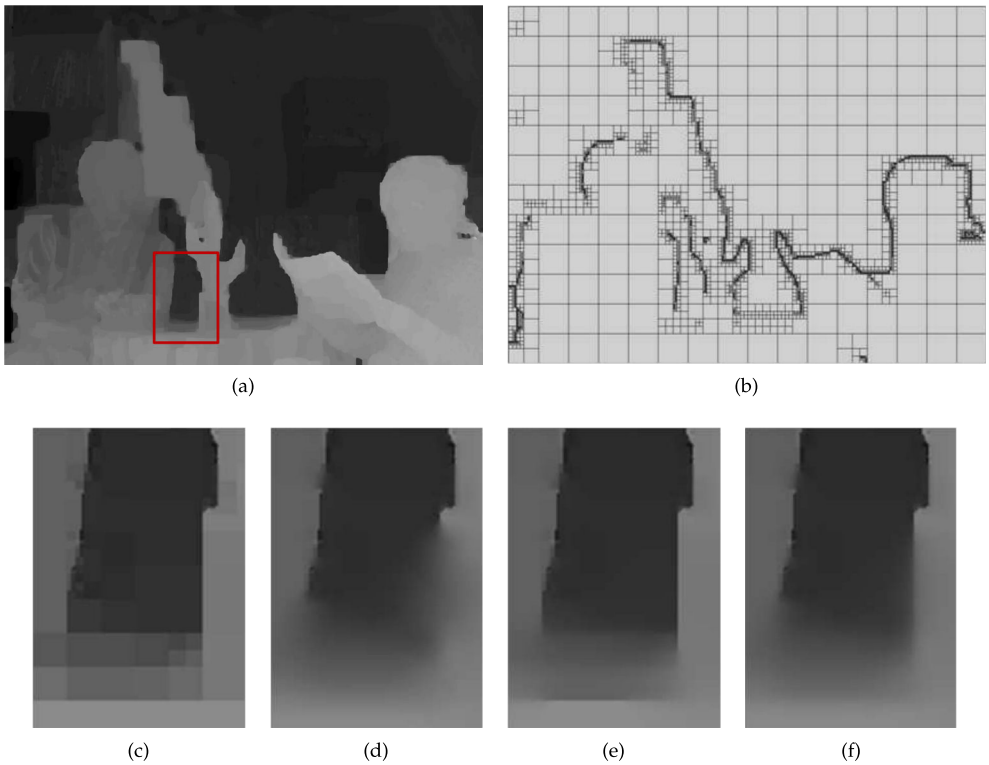
$$Moyenne\_Pondérée = \frac{w_A * A + w_B * B + w_C * C + w_D * D}{w_A + w_B + w_C + w_D} \tag{15}$$

Les FIGURES 76 et 77 illustrent certains exemples des cartes de profondeur codées avec le LAR avec interpolation classique et adaptative. Plus le seuil Th_{PT} est grand, plus l'effet d'interpolation est grand et plus les zones sont lissées, et inversement. Ainsi, pour $Th_{PT} = 0$, il n'y a pas d'interpolation et il y a toujours l'effet de bloc.

Nous remarquons que le post-traitement adaptatif mène à une interpolation plus homogène aux alentours des objets, donnant ainsi une meilleure qualité visuelle.

En résumé, à la sortie du schéma de codage à basse résolution, nous avons :

- la grille de profondeur simple $Grille_{Prof}$, formée de grands blocs dans les zones homogènes de la carte de profondeur et de petits blocs sur les limites des différents plans de profondeur,
- la profondeur codée sur la base de la $Grille_{Prof}$. La carte de profondeur reconstruite garde ainsi finement les contours de la profondeur. Ceci permet une haute qualité de synthèse de vue intermédiaire, comme montré dans la Section 4.5,
- la texture codée aussi sur la base de la $Grille_{Prof}$. La texture reconstruite est ainsi de basse résolution et ne contient pas les détails des différents objets dans la scène 2D.



(a) (b)

(c) (d) (e) (f)

FIGURE 76. Carte de profondeur de Newspaper vue 6 image 1 codée avec le LAR classique à 0.03 bpp avec $\{Q_p = 57;\ Th_{Quad} = 38\}$ (a) originale ; (b) grille ; (c) avec le LAR sans interpolation (PSNR = 35.17 dB) ; (d) avec interpolation classique (PSNR = 35.83 dB) ; (e) avec interpolation adaptative $Th_{PT} = Th_{Quad}$ (PSNR = 36.32 dB) ; (f) avec interpolation adaptative $Th_{PT} = 2 * Th_{Quad}$ (PSNR = 36.21 dB).

4.5 EXPÉRIMENTATIONS ET RÉSULTATS SUR LE CODAGE DE LA PROFONDEUR

L'approche proposée est testée sur les séquences 3D de référence fournies par MPEG (voir FIGURE 78) (images réelles : *Balloons*, *BookArrival*, *Kendo* et *Newspaper* de taille 1024x768, et images de synthèse par ordinateur : *GTFly* et *UndoDancer* de très haute définition (1920x1080).). Les cartes de profondeur associées à ces séquences sont caractérisées par 1) les gradients faibles notamment sur la terre et 2) les gradients forts entre les objets du premier plan et ceux du fond. Cette diversité de gradients nous aide à vérifier la performance de la technique d'interpolation adaptative. En outre, ces gradients ont une multitude de directions permettant de vérifier la performance de la technique du "Meilleur Prédicteur".

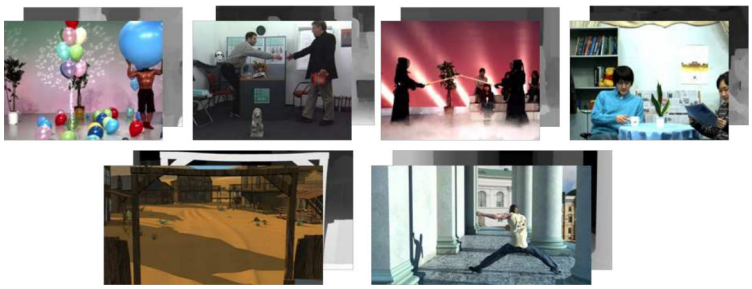
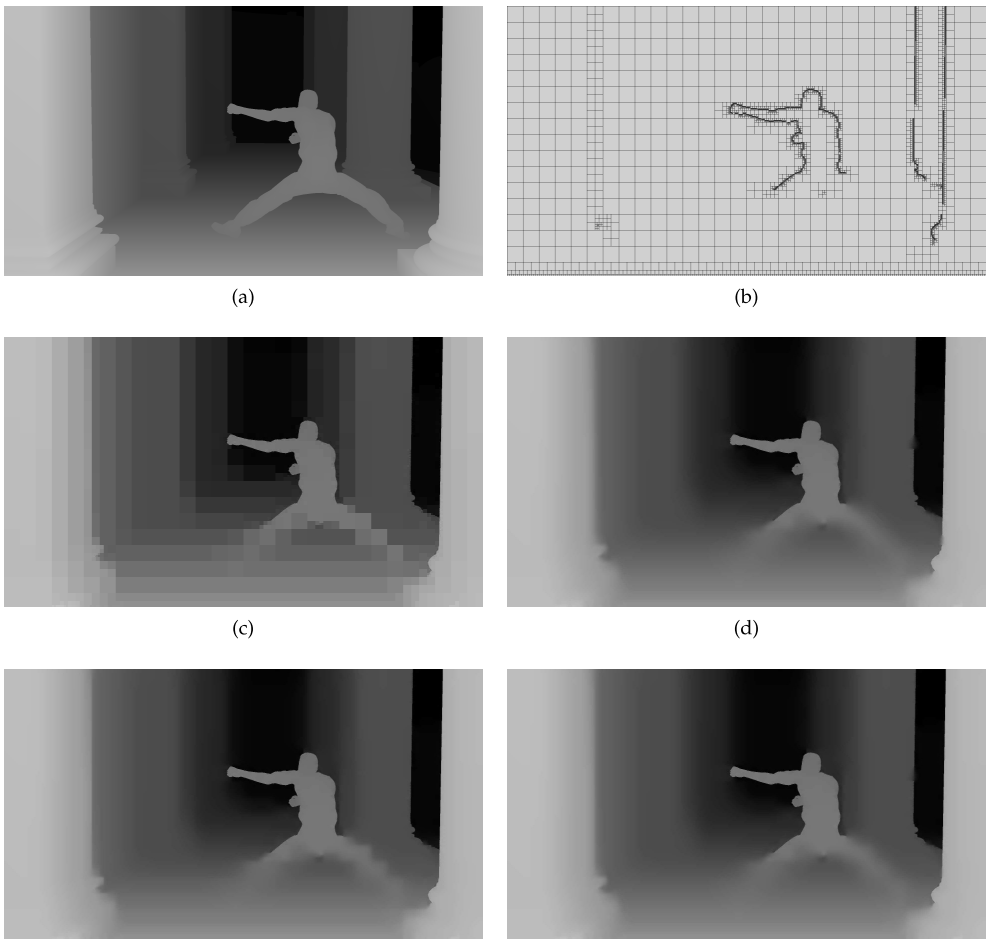


FIGURE 78. Images de références de MPEG 3D.

Les résultats sont comparés avec les codeurs d'image de l'état de l'art de moyenne complexité, tels que : JPEG et JPEGXR (logiciel de référence 1.41) pour la compression avec perte, et JPEGLS (version 0.6.4.1) pour la compression sans perte. Ce choix est motivé par le fait que :



(a) (b) (c) (d) (e) (f)

FIGURE 77. Carte de profondeur de UndoDancer vue 1 image 250 codée avec le LAR classique à 0.12 bpp avec $\{Q_p = 70;\ Th_{Quad} = 47\}$ (a) originale ; (b) grille ; (c) sans interpolation (PSNR = 34.93 dB) ; (d) avec interpolation classique (PSNR = 33.44 dB) ; (e) avec interpolation adaptative $Th_{PT} = Th_{Quad}$ (PSNR = 35.57 dB) ; (f) avec interpolation adaptative $Th_{PT} = 2 * Th_{Quad}$ (PSNR = 35.27 dB).

- JPEG est le référence de codage d'image à faible complexité et reste très largement utilisé.
- JPEXR possède approximativement la même complexité que le LAR, et offre une compression avec et sans perte, mais il n'offre que deux niveaux de scalabilité.
- JPEG-LS est optimisé pour le codage sans perte uniquement.

Le JPEG2000 et le 3DHEVC sont exclus des expérimentations, étant de grande complexité par rapport au LAR. La comparaison n'est pas ainsi faisable.

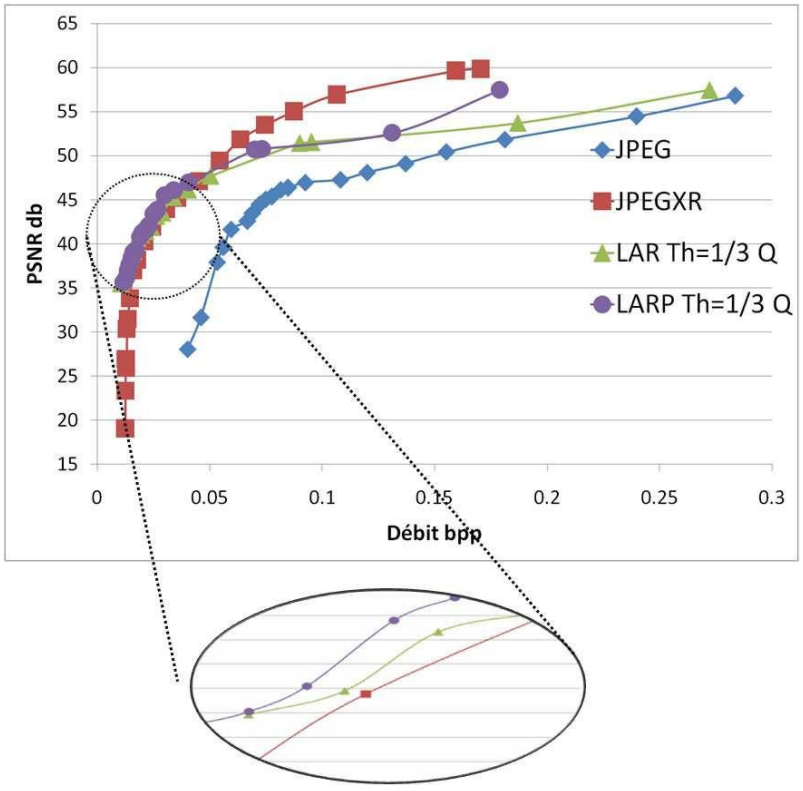
4.5.1 *Résultats objectifs sur les cartes de profondeur*

Dans une première série d'expériences, des tests objectifs sont effectués pour le codage des cartes de profondeur. La quantité de distorsion de la carte de profondeur reconstruite, peut être contrôlée par le choix du seuil d'homogénéité Th_{Quad} et le pas de quantification Q_p. Les résultats débit-distorsion de l'approche proposée, sont ainsi générés en variant le pas de quantification Q_p de 1 (pas de quantification) jusqu'à 120, et en saisissant initialement le seuil $Th_{Quad} = \frac{1}{3}Q_p$, puis $Th_Quad = \frac{2}{3}Q_p$, (voir FIGURE 79 et 80). Pour $Th_{Quad} = \frac{2}{3}Q_p$, la grille de profondeur $Grille_{prof}$ contient de larges blocs en comparaison avec $Th_{quad} = \frac{1}{3}Q_p$, mais la quantification utilisée est plus fine.

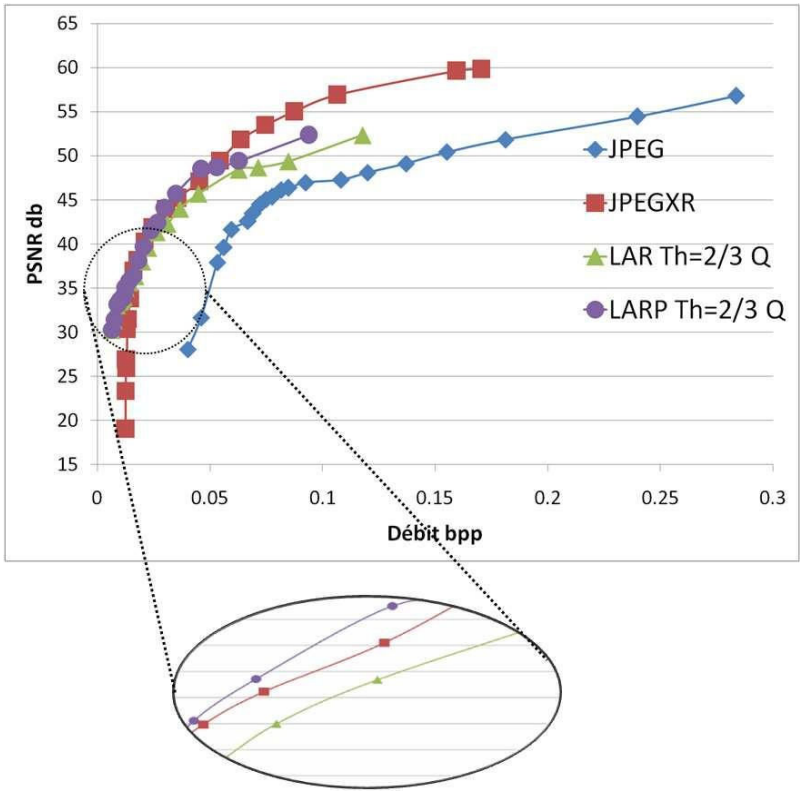
La compression des cartes de profondeur avec l'approche proposée (LARP), permet un gain en PSNR de 1 à 2 dB suivant le débit, avec 20 à 30% de gain en débit par rapport au LAR classique (codage indépendant).

En comparant, les performances avec celles de JPEG et JPEGXR pour la compression avec pertes, il apparaît :

- un gain significatif quelque soient les débits par rapport à JPEG.
- un gain important jusqu'à 50 % à bas débit (< 0.05 bpp) par rapport à JPEGXR, comme le montrent les images zoomées.

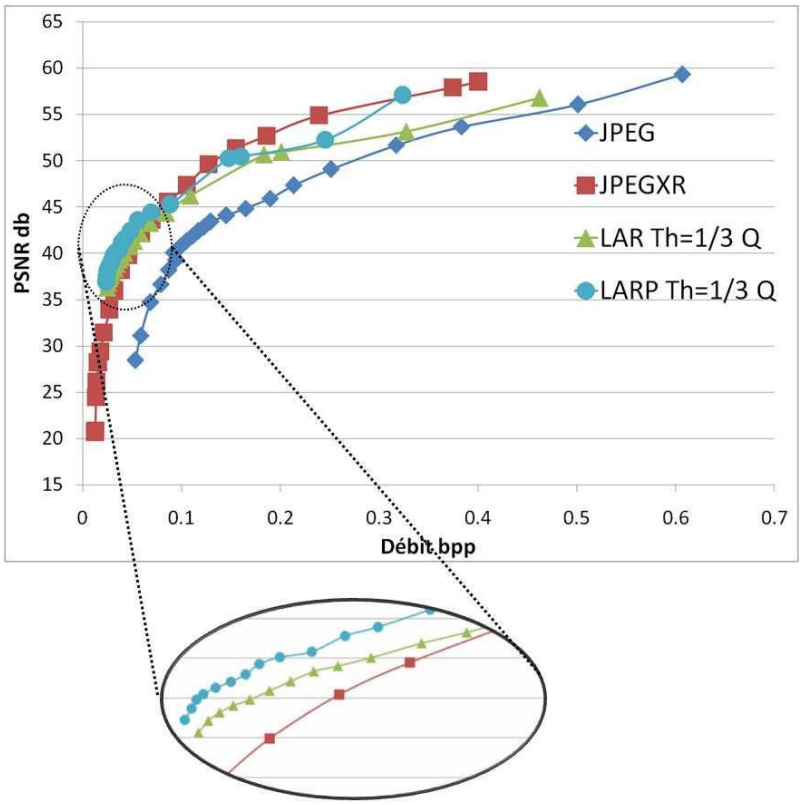


(a) $\mathrm{Th}_{\mathrm{Quad}} = \frac{1}{3}\mathrm{Q_p}$

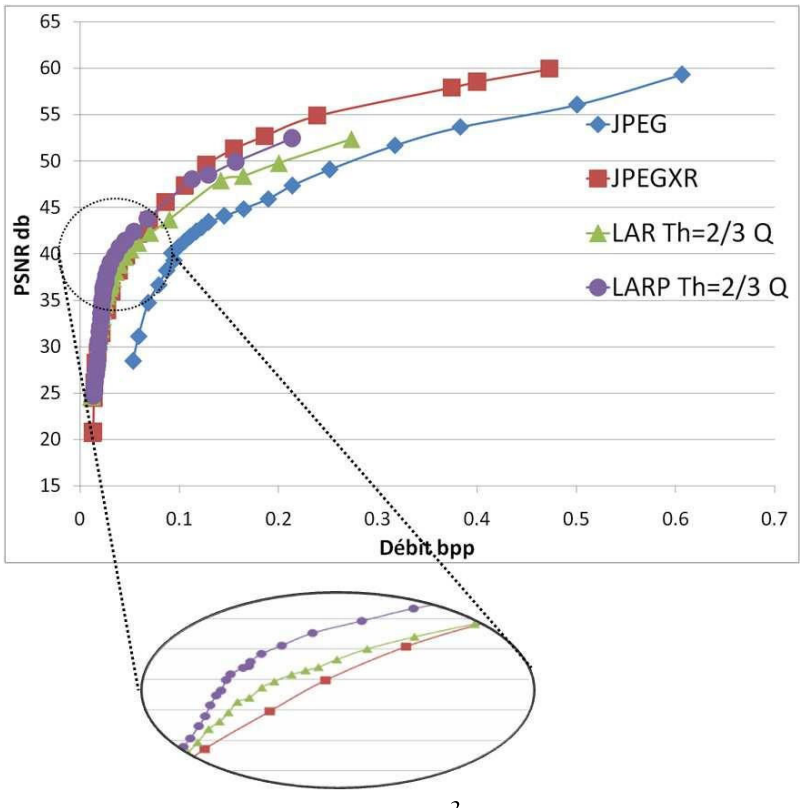


(b) $\mathrm{Th}_{\mathrm{Quad}} = \frac{2}{3}\mathrm{Q_p}$

FIGURE 79. Courbes débits-distorsion de la carte de profondeur de UndoDancer image 250 vue 1.



(a) $\mathrm{Th}_{\mathrm{Quad}} = \frac{1}{3}\mathrm{Q_p}$



(b) $\mathrm{Th}_{\mathrm{Quad}} = \frac{2}{3}\mathrm{Q_p}$

FIGURE 80. Courbes débits-distorsion de la carte de profondeur de GTFly image 157 vue 1.

Pour des images de haute qualité visant des applications telles que la FTV ou la 3DTV, la compression sans perte peut être préférée. Les performances de codage ont été ainsi comparées avec le standard de compression sans perte JPEGLS, le mode "lossless" de JPEGXR. Comme le montre le tableau 1, l'outil proposé atteint un gain de compression sans perte plus élevé que le LAR classique (33% pour les images réelles et 21% pour les images de synthèse) et que JPEGXR (59% pour les images réelles et 64% pour les images de synthèse). Par contre, on peut constater que le codeur JPEGLS obtient de meilleurs résultats que LARP. En effet, le JPEG-LS est construit et optimisé juste pour le codage sans perte. Par contre, le LAR est dédié pour le codage avec et sans perte.

Table 1. Débit en bpp des cartes de profondeur codées sans perte

Carte Profondeur	Débit (bpp)			
	LAR	LARP	JPEGXR	JPEGLS
GTFly	0.84	0.62	1.70	0.33
Balloons	1.10	0.73	1.85	0.44
Newspaper	1.21	0.89	1.97	0.52
BookArrival	1.45	0.92	2.36	0.50
UndoDancer	0.52	0.44	1.25	0.43
Moyenne	1.31	0.72	1.82	0.44

4.5.2 *Résultats Visuels sur les cartes de profondeur*

Même lorsque le PSNR des cartes reconstruites par JPEGXR est plus élevé que celui des cartes reconstruites par la méthode proposée, nous pouvons remarquer que le LARP améliore la qualité visuelle des cartes de profondeur reconstruites, en comparaison avec le LAR classique et JPEGXR (voir FIGURE 81, 82).

D'une part, dans les FIGURES 81.c et 82.c, nous remarquons une meilleure qualité visuelle par rapport aux FIGURES 81.b, 82.b et 81.e et 82.e , respectivement. En effet, la technique de "Meilleur Prédicteur" permet une meilleure prédiction des valeurs de profondeur, ce qui minimise les distorsions notamment au niveau des contours des objets.

D'autre part, les FIGURES 81.d et 82.d présentent la meilleure qualité. L'interpolation adaptative appliquée après la technique du "Meilleur Prédicteur", permet de lisser la carte de profondeur tout en gardant un gradient fort sur les contours.

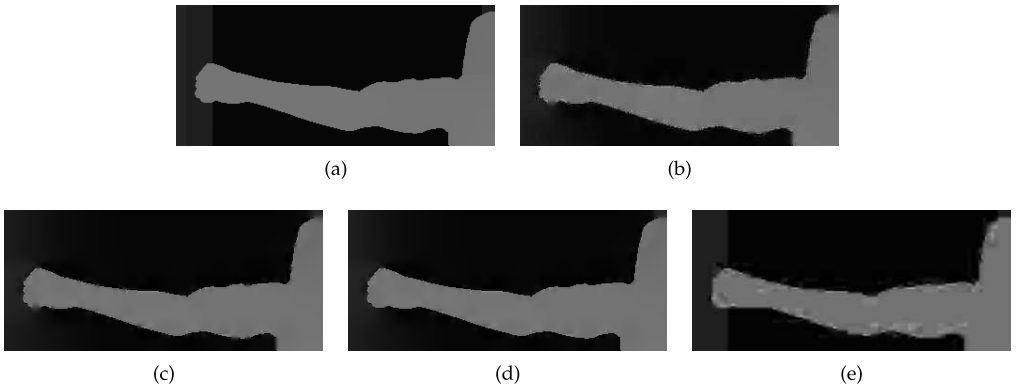
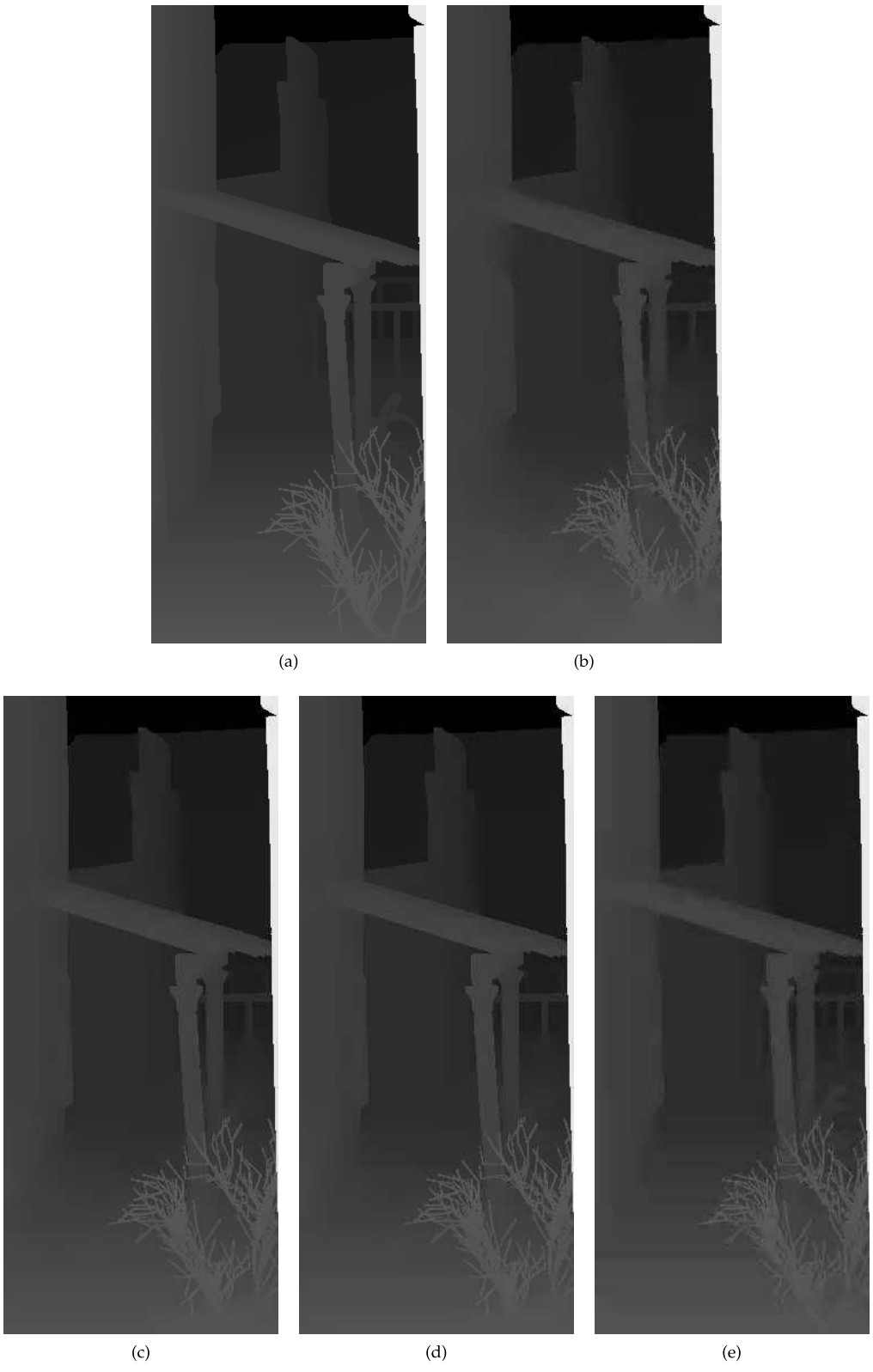


Figure 81. Comparaison de la qualité visuelle de la carte de profondeur reconstruite de UndoDancer vue 1 image 250 à 0.006 bpp : (a) originale ; (b) codée avec le LAR classique (PSNR = 30.16 dB, partition initiale de 2638 blocs) ; (c) codée avec l'approche "Meilleur Prédiction" suivie d'une interpolation classique (PSNR = 30.20 dB, partition initiale de 2638 blocs) ; (d) codée avec l'approche "Meilleur Prédiction" suivie d'une interpolation adaptative (PSNR = 30.28 dB, partition initiale de 2638 blocs) ; (d) carte de profondeur codée avec JPEGXR (PSNR = 28 dB).



(a) (b)

(c) (d) (e)

FIGURE 82. Comparaison de la qualité visuelle de la carte de profondeur reconstruite de GTFly vue 1 image 157 à 0.08 bpp : (a) originale ; (b) codée avec le LAR classique (PSNR = 43.76 dB, partition initiale de 20227 blocs) ; (c) codée avec l'approche "Meilleur Prédiction" suivie d'une interpolation classique (PSNR = 44.65 dB, partition initiale de 25436 blocs) ; (d) codée avec l'approche "Meilleur Prédiction" suivie d'une interpolation adaptative (PSNR = 44.92 dB, partition initiale de 25436 blocs) ; (e) codée avec JPEGXR (PSNR = 43 dB).

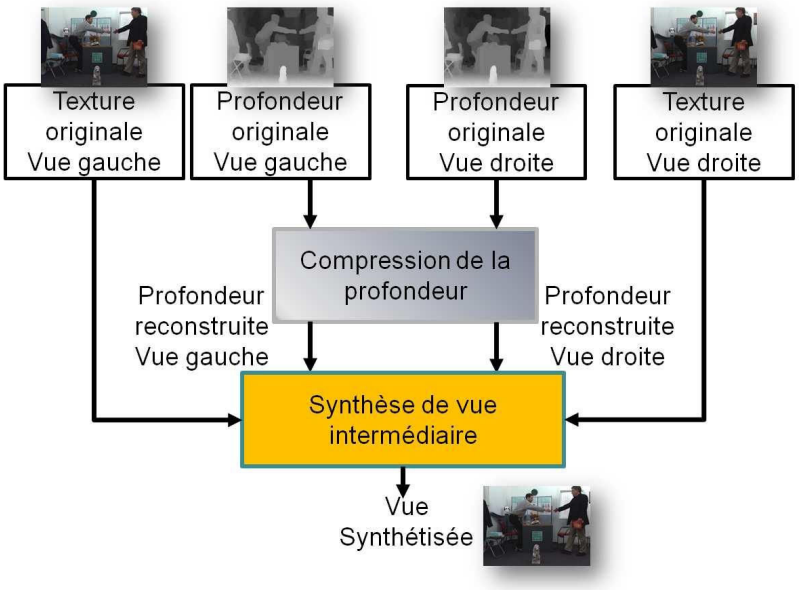
4.5.3 *Résultats visuels sur les vues synthétisées*

Comme mentionné précédemment, l'évaluation réelle de la qualité des images dans un contexte de codage 3D, doit être réalisée sur les images synthétisées. Pour la synthèse des vues intermédiaires, nous utilisons le logiciel *View Synthesis Reference Software (VSRS 3.0)* [26]. Afin d'évaluer l'effet de compression de la profondeur sur la vue synthétisée, dans cette série d'expérience, nous considérons les images de texture originales et les cartes de profondeur compressées avec $Th_{Quad} = \frac{2}{3}Q_p$, (voir FIGURE 83).

Les FIGURES 84, 85 et 86 illustrent les résultats de synthèse de vue : (a) à partir des cartes de profondeur originales ; (b) à partir des cartes compressées avec le LAR classique ; (c) à partir des cartes compressées avec l'outil "Meilleur Prédiction" suivi d'une interpolation classique ; (d) à partir des cartes compressées avec l'outil "Meilleur Prédiction" suivi d'une interpolation adaptative (LARP) ; (e) à partir des cartes compressées avec JPEGXR, respectivement.

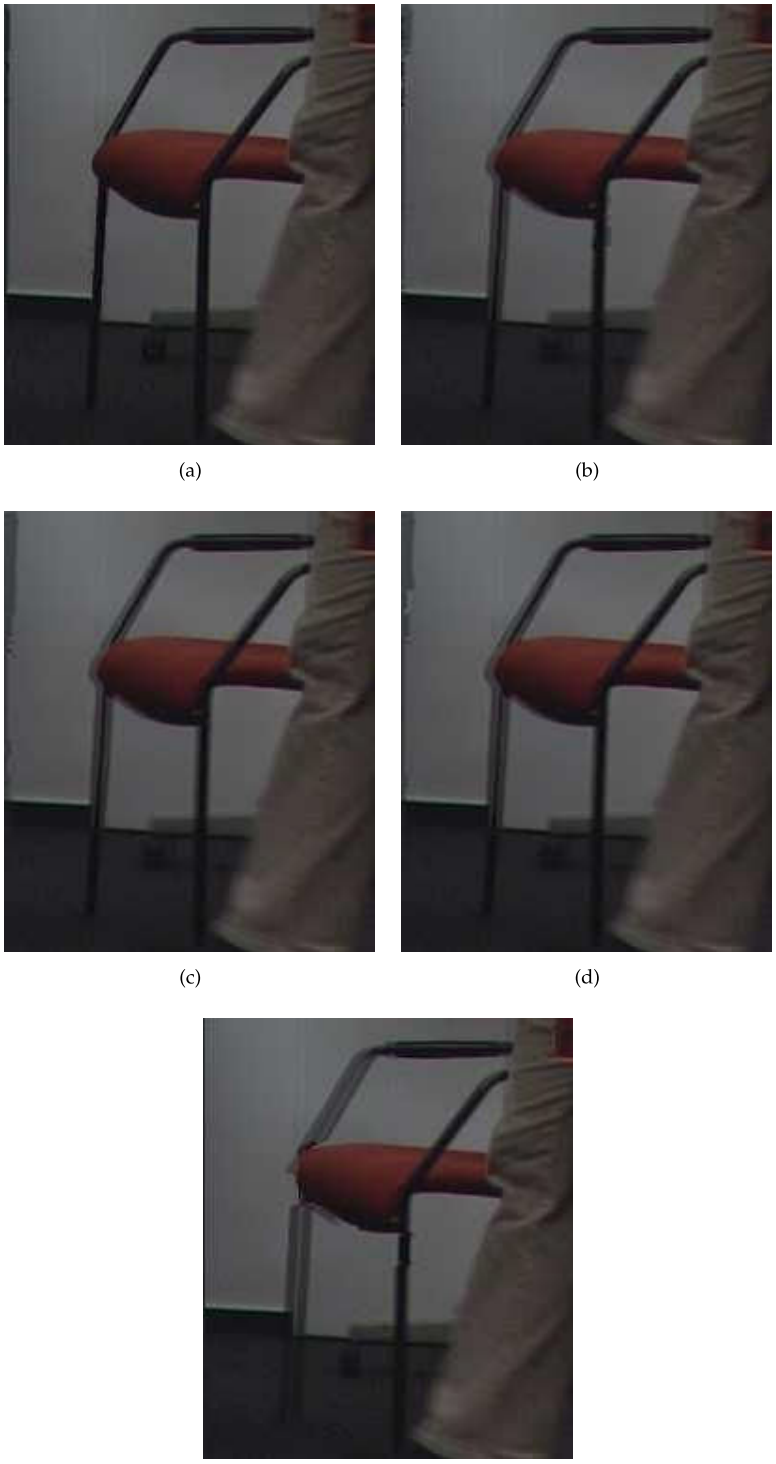
En regardant le bras et le pieds de la chaise sur la FIGURE 84, les contours des ballons sur la FIGURE 85 et le mur et le poteau sur la FIGURE 86, nous remarquons que la qualité visuelle des vues synthétisées à partir des cartes de profondeur compressées en utilisant l'outil proposé, est nettement supérieure que celles des vues synthétisées à partir des cartes de profondeur reconstruites en utilisant le LAR classique ou JPEGXR, notamment sur les contours des objets. La corrélation entre la profondeur et la texture utilisée pour améliorer la prédiction de la profondeur, suivie d'une interpolation adaptative, permet d'augmenter la qualité de la vue synthétisée.

En effet, à bas débit, où il y a de grandes quantifications, le LAR classique et JPEGXR induisent de grandes distorsions. Par contre, la technique du "Meilleur Prédicteur" permet de trouver la direction la plus adéquate du contour local. Ceci permet ainsi, malgré les grandes quantifications, de conserver les directions des contours des objets et de garder la sémantique de la scène. Ceci assure une cohérence entre la texture et la profondeur des objets dans la scène lors du codage de la profondeur et garantit une meilleure qualité visuelle des vues synthétisées.



(a)

FIGURE 83. Schéma de synthèse de vues intermédiaires utilisé dans l'expérimentation.



(a) (b)

(c) (d)

(e)

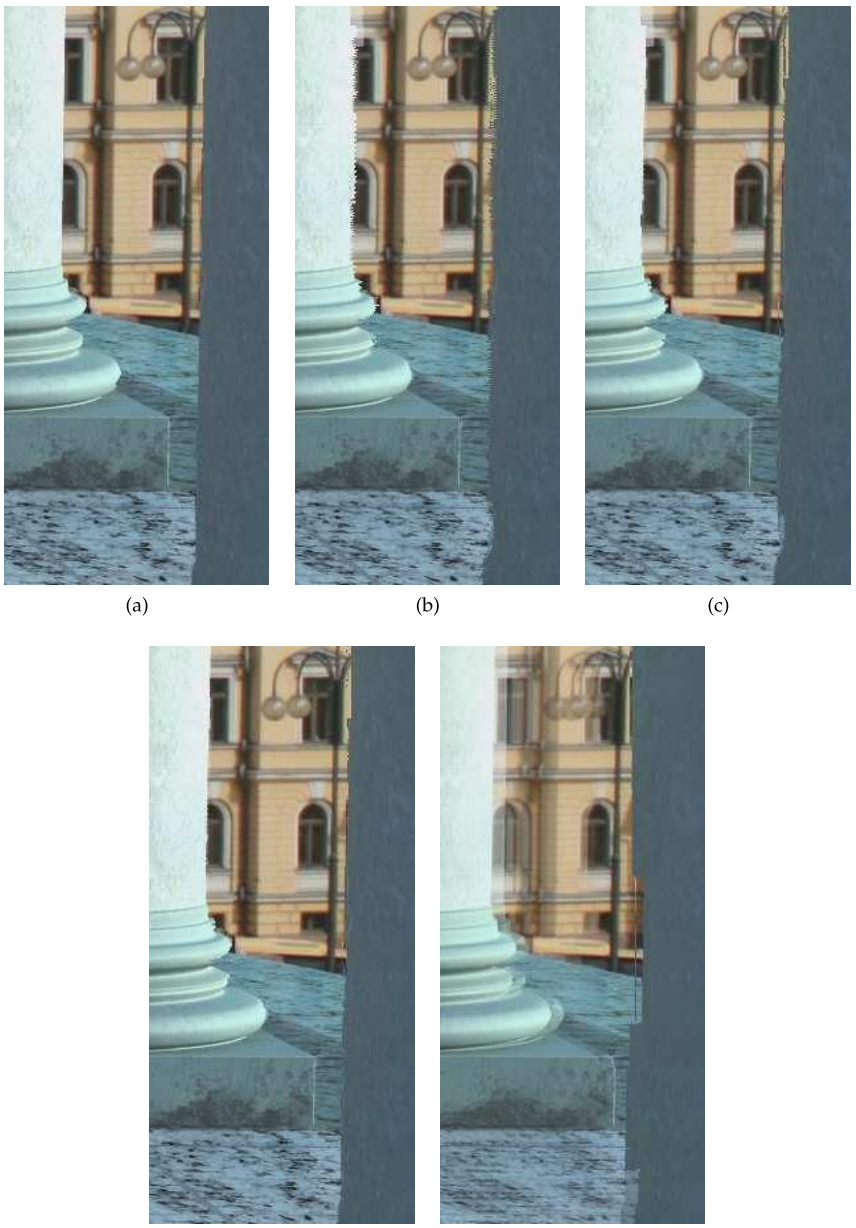
FIGURE 84. Comparaison de la qualité visuelle de la vue synthétisée BookArrival vue 9 image 033 à 0.012 bpp utilisant les cartes de profondeur (a) originales ; (b)reconstruites avec le LAR classique (PSNR de la profondeur = 26.6 dB, partition initiale de 1663 blocs) ; (c) reconstruites avec l'outil "Meilleur Prédiction" suivi d'une interpolation classique (PSNR de la profondeur = 25.7 dB, partition initiale de 1154 blocs) ; (d) reconstruites avec l'outil "Meilleur Prédiction" suivi d'une interpolation adaptative (PSNR de la profondeur = 26 dB, partition initiale de 1154 blocs) ; e) reconstruites avec JPEGXR (PSNR de la profondeur = 21.5 dB).

(a) (b)

(c) (d)

(e)

FIGURE 85. Comparaison de la qualité visuelle de la vue synthétisée Balloons vue 4 image 1 à 0.013 bpp utilisant les cartes de profondeur (a) originales ; (b)reconstruites avec le LAR classique (PSNR de la profondeur = 29.56 dB, partition initiale de 1894 blocs) ; (c) reconstruites avec l'outil "Meilleur Prédiction" suivi d'une interpolation classique (PSNR de la profondeur = 28.77 dB, partition initiale de 1450 blocs) ; (d) reconstruites avec l'outil "Meilleur Prédiction" suivie d'une interpolation adaptative (PSNR de la profondeur = 29 dB, partition initiale de 1450 blocs) ; (e) reconstruites avec JPEGXR (PSNR de la profondeur = 22.65 dB).



(a) (b) (c)

(d) (e)

FIGURE 86. Comparaison de la qualité visuelle de la vue synthétisée UndoDancer vue 3 image 250 à 0.012 bpp utilisant les cartes de profondeur (a) originales ; (b)reconstruites avec le LAR classique (PSNR de la profondeur = 34.99 dB, partition initiale de 4867 blocs) ; (c) reconstruites avec l'outil "Meilleur Prédiction" suivi d'une interpolation classique (PSNR de la profondeur = 35.08 dB, partition initiale de 4867 blocs) ; (d) reconstruites avec l'outil "Meilleur Prédiction" suivie d'une interpolation adaptative (PSNR de la profondeur = 37 dB, partition initiale de 4867 blocs) ; (e) reconstruites avec JPEGXR (PSNR de la profondeur = 19.03 dB).

Comme présenté dans la Section 4.3, la deuxième étape du schéma de codage 2D+Z scalable que nous proposons, est le codage à haute résolution de la texture. Cette étape permet de raffiner la qualité de la texture afin de pouvoir récupérer ces détails (voir FIGURE 67). Dans les sous-sections suivantes, nous détaillons la procédure de raffinement de qualité, et analysons le surcoût en terme de débit d'un tel schéma de codage scalable.

4.6.1 *Rehaussement de la qualité de la texture*

Comme mentionné précédemment (voir Section 4.3), la résolution de l'image reconstruite après codage par le LAR est directement liée à la grille du QuadTree. La texture à haute résolution doit être ainsi codée avec une grille plus fine que celle de la profondeur afin de pouvoir récupérer les détails de la texture. L'étape de raffinement de la qualité de la texture est réalisée en deux parties (voir FIGURE 87) :

1) Raffinement de la grille : le raffinement de la $\mathrm{Grille}_{\mathrm{Prof}}$ consiste à y ajouter l'information de texture. Tout d'abord, une grille, $\mathrm{Grille}_{\mathrm{Tex}}$, est calculée à partir des composantes couleur de la texture (Y, C_b, C_r) par la même technique de partitionnement QuadTree expliquée dans la Section 4.2, et avec le même seuil $\mathrm{Th}_{\mathrm{Quad}}$ utilisé pour la $\mathrm{Grille}_{\mathrm{Prof}}$ dans le schéma de codage à basse résolution. Ensuite, la taille de chaque bloc dans la grille raffinée $\mathrm{Grille}_{\mathrm{Prof+Tex}}$ correspond au minimum des tailles du bloc entre $\mathrm{Grille}_{\mathrm{Prof}}$ et $\mathrm{Grille}_{\mathrm{Tex}}$. La texture contenant plus de détails de couleur par rapport à la profondeur, la $\mathrm{Grille}_{\mathrm{Prof+Tex}}$ obtenue est ainsi raffinée avec plus de petits blocs.
2) Codage multi-résolution des blocs décomposés : la $\mathrm{Grille}_{\mathrm{Prof}}$ étant raffinée, certains blocs vont être décomposés suivant la $\mathrm{Grille}_{\mathrm{Prof+Tex}}$. À chaque niveau de la pyramide et suivant la grille raffinée $\mathrm{Grille}_{\mathrm{Prof+Tex}}$, seuls les blocs décomposés suivant la $\mathrm{Grille}_{\mathrm{Prof+Tex}}$ (non décomposés suivant la $\mathrm{Grille}_{\mathrm{Prof}}$) sont codés par le schéma de multi-résolution 2D du LAR expliqué dans la Section 4.2. La FIGURE 88, illustre un exemple de raffinement de la grille de profondeur et de la texture, où seuls les blocs en pointillés sont codés par le LAR 2D. Le facteur de quantification Q_p à utiliser dans le schéma de raffinement est indépendant du celui utilisé dans le schéma à basse résolution. Toutefois, pour garder l'homogénéité de la qualité entre la profondeur codée et la texture raffinée, nous utilisons en général la même valeur du facteur de quantification. La texture codée suivant la grille raffinée contient ainsi plus de détails sur les couleurs de la scène. C'est la texture à haute résolution.

Les informations fournies à cette étape du schéma scalable proposé (codage à haute résolution) sont ainsi :

- le raffinement de la grille, appelé Δ_{Grille}, et qui est la différence entre les deux grilles $\mathrm{Grille}_{\mathrm{Prof+Tex}}$ et $\mathrm{Grille}_{\mathrm{Prof}}$.
- les erreurs de prédiction des blocs raffinés.

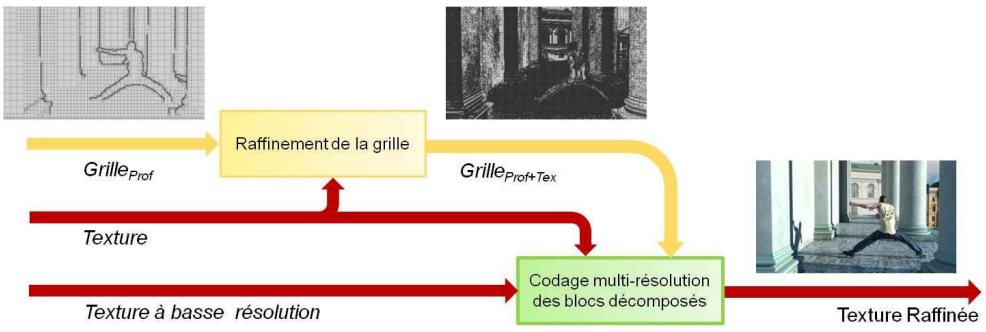
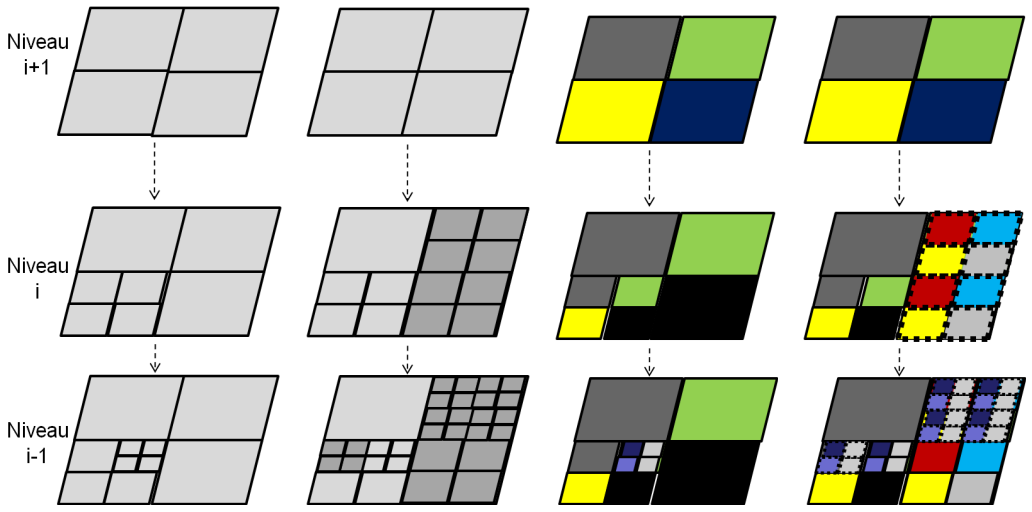


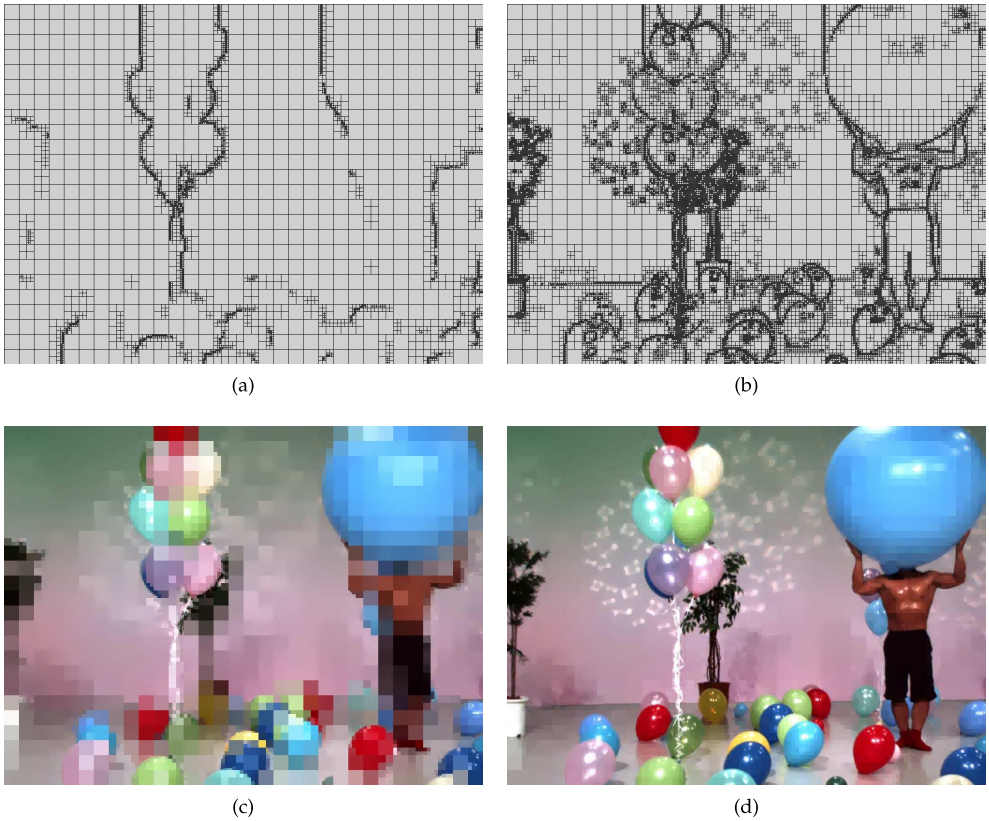
FIGURE 87. Deuxième étape du codage scalable proposé : codage à haute résolution de la texture.

Les FIGURES 89 et 90 illustrent quelques exemples de raffinement d'images de texture.



(a) Grille profondeur (b) Grille profondeur plus texture (c) Texture basse résolution basee grille profondeur (d) Texture raffinée basée grille profondeur plus Texture

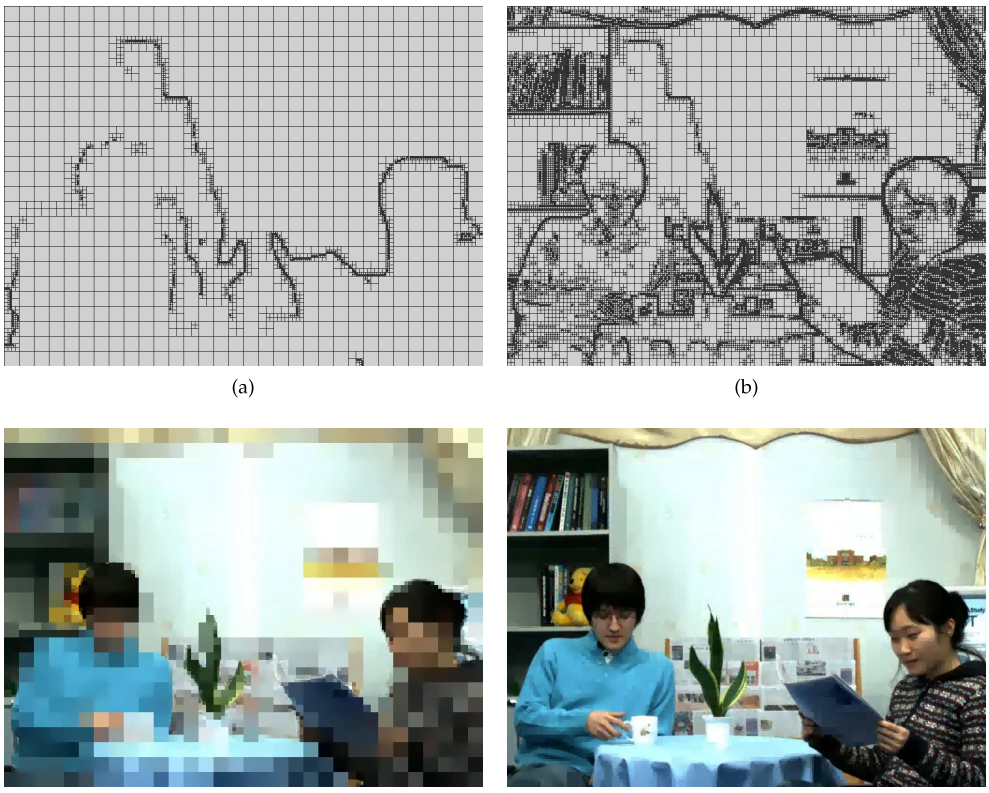
FIGURE 88. Codage multi-résolution des blocs décomposés : c) Découpage et Prédiction des blocs de la texture suivant la grille profondeur ; (d) Raffinage de la texture basse résolution suivant la grille profondeur plus texture et Prédiction des blocs raffinés (en pointillés).



(a) (b)

(c) (d)

FIGURE 89. Exemple de raffinement de Balloons vue 5 image 1, $\{Q_p = 50; Th_{Quad} = 33\}$: (a) grille profondeur ; (b) grille profondeur + texture ; (c) texture basse résolution (0.05 bpp, PSNR = 20.29 dB) ; (d) texture raffinée (0.17 bpp, PSNR = 32.7 dB).



(a) (b)

(c) (d)

FIGURE 90. Exemple de raffinement de Newspaper vue 6 image 1, $\{Q_p = 70; Th_{Quad} = 46\}$: (a) grille profondeur ; (b) grille profondeur + texture ; (c) texture basse résolution (0.04 bpp, PSNR = 18.26 dB) ; (d) texture raffinée (0.17 bpp, PSNR = 30 dB).

4.6.2 *Étude du coût du schéma de codage joint et scalable*

Le schéma proposé étant **joint** et **scalable**, l'étude de son coût nécessite une comparaison avec le coût du schéma de codage non joint d'une part et non scalable d'autre part (voir FIGURE 91). Une telle étude permet de déterminer le surcoût apporté par le schéma proposé.

a) D'une part, un schéma de codage indépendant (non joint) basé sur le LAR fournit (voir FIGURE 91.a) :

- la partition *Quadtree* de la profondeur ($Grille_{Prof}$),
- la carte de profondeur codée suivant la $Grille_{Prof}$ (Prof),
- la grille de la texture ($Grille_{Tex}$),
- la texture codée suivant sa propre grille $Grille_{Tex}$ (Tex_{HR}).

Le coût de grille ($Coût_{Grille}$) et le coût de texture ($Coût_{Tex}$) du schéma de codage indépendant, peuvent être ainsi traduits par les équations (16) et (17), respectivement.

$$Coût_{Grille} = Coût_{Grille_{Prof}} + Coût_{Grille_{Tex}} \quad (16)$$

$$Coût_{Tex} = Coût_{Tex_{HR}} \quad (17)$$

b) Par contre, un schéma de codage joint, code la texture et la profondeur conjointement suivant la grille basée texture et profondeur ($Grille_{Prof+Tex}$) (voir FIGURE 91.b). Le coût de la grille d'un tel schéma est traduit par l'équation (18).

$$Coût_{Grille} = Coût_{Grille_{Prof+Tex}} \quad (18)$$

Or, un bloc décomposé dans la grille de profondeur peut même être décomposé dans la grille de la texture, grâce à la corrélation entre la texture et la profondeur. Le coût de la grille profondeur + texture va ainsi être inférieur au coût de la grille profondeur ajouté au coût de la grille de la texture (Eq. 19).

$$\text{Coût}_{\text{Grille}_{\text{Prof+Tex}}} \leqslant \text{Coût}_{\text{Grille}_{\text{Prof}}} + \text{Coût}_{\text{Grille}_{\text{Tex}}} \tag{19}$$

c) D'autre part, un schéma non scalable (voir FIGURE 91.c) fournit :

- la partition *Quadtree* de la profondeur + texture ($\text{Grille}_{\text{Prof+Tex}}$),
- la carte de profondeur codée suivant la $\text{Grille}_{\text{Prof+Tex}}$ (Prof),
- la texture codée suivant la grille $\text{Grille}_{\text{Prof+Tex}}$ (Tex_{HR}).

Le coût de la grille et le coût de la texture sont ainsi exprimés par les équations (20) et (21).

$$\begin{aligned} \text{Coût}_{\text{Grille}} &= \text{Coût}_{\text{Grille}_{\text{Prof+Tex}}} && (20)\\ \text{Coût}_{\text{Tex}} &= \text{Coût}_{\text{Tex}_{\text{HR}}} && (21) \end{aligned}$$

d) Alors que le schéma scalable proposé fournit (voir FIGURE 91.d) :

- la $\text{Grille}_{\text{Prof}}$,
- la carte de profondeur codée suivant cette grille (Prof),
- la texture à basse résolution (Tex_{BR}),
- le raffinement de la $\text{Grille}_{\text{Prof}}$ (Δ_{Grille}),
- le raffinement de la texture (Δ_{Tex}).

Le coût de grille ($\text{Coût}_{\text{Grille}}$) et le coût de la texture (Coût_{Tex}) du schéma de codage scalable sont ainsi exprimés par les équations (22) et (23), respectivement.

$$\begin{aligned} \text{Coût}_{\text{Grille}} &= \text{Coût}_{\text{Grille}_{\text{Prof}}} + \text{Coût}_{\Delta_{\text{Grille}}} && (22)\\ \text{Coût}_{\text{Tex}} &= \text{Coût}_{\text{Tex}_{\text{BR}}} + \text{Coût}_{\Delta_{\text{Tex}}} && (23) \end{aligned}$$

$$\text{Coût}_{\text{Grille}_{\text{Prof}}} + \text{Coût}_{\Delta_{\text{Grille}}} \geqslant \text{Coût}_{\text{Grille}_{\text{Tex+Prof}}} \tag{24}$$

$$\text{Coût}_{\text{TexBR}} + \text{Coût}_{\Delta_{\text{Tex}}} > \text{Coût}_{\text{TexHR}} \tag{25}$$

En effet, considérons un bloc dans la grille de profondeur décomposé jusqu'à un certain niveau l. Cette grille est ensuite raffinée en fonction de la texture. La décomposition d'un tel bloc va donc être continuée à partir du niveau l vers un niveau plus inférieur. L'information de décomposition d'un tel bloc est ainsi envoyée deux fois. Par contre, le même bloc dans la grille profondeur + texture, va être décomposé directement jusqu'au niveau le plus inférieur. L'information de décomposition n'est ainsi envoyée dans ce cas là qu'une seule fois, d'où l'inégalité (25), où le coût de codage de la grille de profondeur ajouté au coût du raffinement de la grille, va être supérieur au coût de la grille profondeur + texture (voir Eq. 24).

D'autre part, en comparaison avec une texture à haute résolution, le codage de la texture à basse résolution induit le phénomène suivant : lors du codage de la texture à basse résolution, la prédiction d'un bloc courant est faite à partir des blocs voisins à basse résolution. Un tel phénomène implique une moins bonne prédiction et donc une grande erreur de prédiction. Le coût de raffinement va ainsi être élevé, d'où l'inégalité (25), où le coût de codage de la texture à basse résolution ajouté au coût du raffinement de la texture, va être supérieur au coût de la texture codée directement à haute résolution (voir Eq. 25).

Un tel schéma scalable engendre donc un surcoût en terme de débit, par rapport au schéma de codage non scalable (voir Eq. (26)).

$$\begin{aligned} \text{Surcoût}_{\text{Grille}} &= \text{Coût}_{\text{Grille}_{\text{Tex+Prof}}} - (\text{Coût}_{\text{Grille}_{\text{Prof}}} + \text{Coût}_{\Delta_{\text{Grille}}}).\\ \text{Surcoût}_{\text{Tex}} &= \text{Coût}_{\text{TexHR}} - (\text{Coût}_{\text{TexBR}} + \text{Coût}_{\Delta_{\text{Tex}}}).\\ \text{Surcoût} &= \text{Surcoût}_{\text{Grille}} + \text{Surcoût}_{\text{Tex}}. && (26) \end{aligned}$$

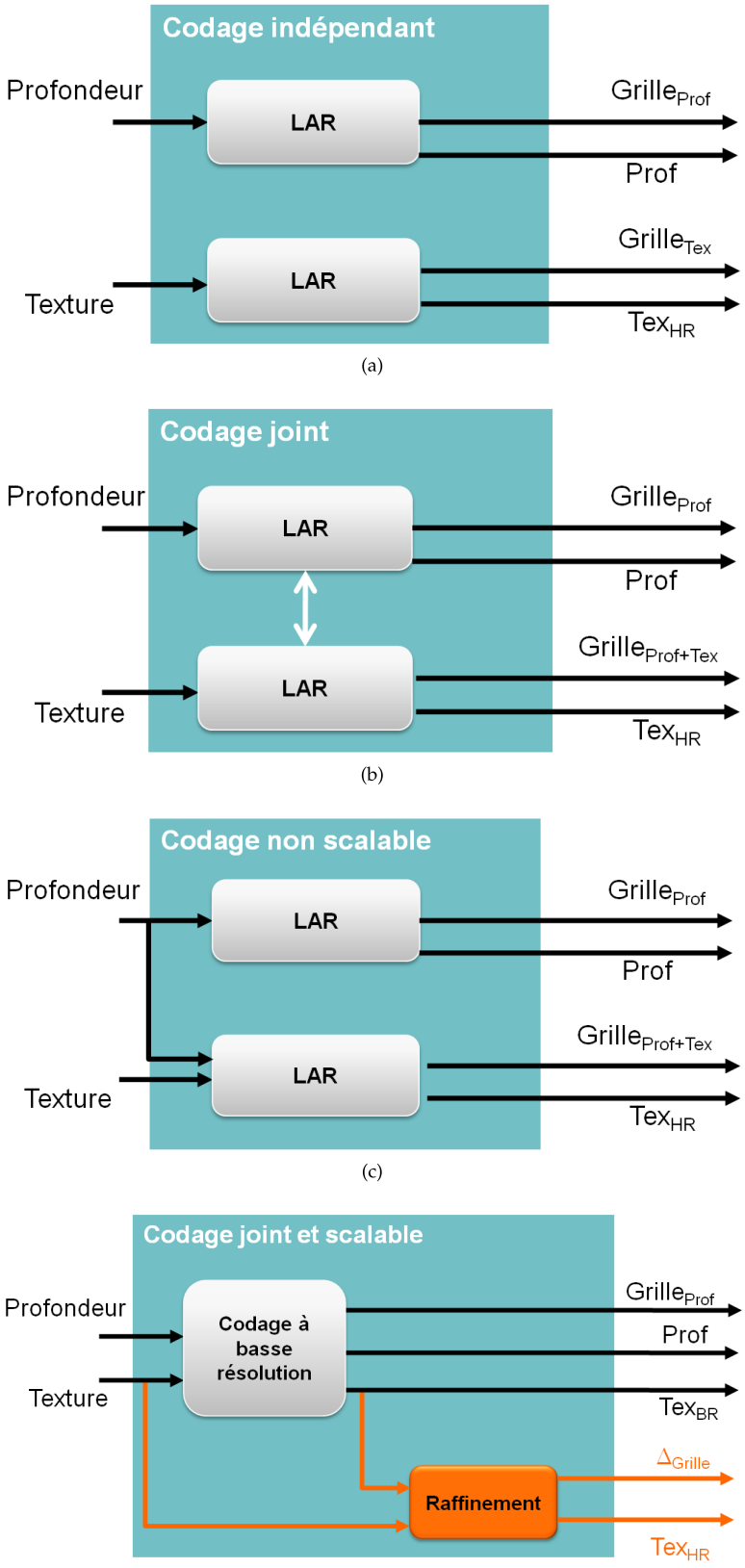


FIGURE 91. a) schéma de codage indépendant (non joint) ; (b) schéma de codage joint ; (c) schéma de codage non scalable ; (d) schéma de codage joint et scalable proposé.

4.6.3 *Résultats de surcoût du schéma de codage scalable proposé*

Le schéma de codage à haute résolution est testé sur les images 3D de référence fournies par MPEG. Le surcoût du schéma de codage scalable est présenté dans les tableaux suivants.

Les tableaux 2 et 4 donnent des exemples de débit-distorsion du schéma de codage non scalable, pour les images Balloons et UndoDancer, pour différents couples $\{Q_p, Th_{Quad}\}$. Ces tableaux génèrent respectivement le coût de la grille de profondeur, le coût de la profondeur, le PSNR de la profondeur, le coût de la grille texture, le coût de la texture et sa qualité.

Les tableaux 3 et 5 donnent deux exemples de débit-distorsion et de surcoût du schéma de codage scalable pour les mêmes images. Ces tableaux génèrent respectivement le coût de la texture basse résolution, le coût de raffinement de la grille, le coût de raffinement de la texture, la qualité de la texture raffinée, et le surcoût engendré par le schéma scalable.

Le surcoût généré par le schéma scalable dépend de chaque image 3D, et plus particulièrement de la carte de profondeur. Si celle ci est homogène, alors dans le schéma de codage à basse résolution, aucun bloc n'est décomposé. La décomposition est par suite faite entièrement durant le schéma de codage à haute résolution. Le codage se ressemble ainsi à un schéma de codage non scalable, et aucun surcoût n'est ainsi généré. Inversement, si la carte de profondeur est à haute activité, des blocs sont décomposés jusqu'à un certain niveau de la pyramide, durant le schéma de codage à basse résolution. Durant le codage à haute résolution, la grille de partitionnement est ensuite raffinée en fonction de la texture, et la décomposition est ainsi reprise. Une nouvelle information de partitionnement est donc renvoyée, générant ainsi le surcoût.

De plus, le tableau 6 donne la moyenne du surcoût pour chacune des images 3D de référence fournies par MPEG. La moyenne de surcoût du schéma de codage scalable par rapport au schéma de codage non scalable sur les images 3D de référence est de l'ordre de 0.05 bpp (15% de surcoût par rapport au débit généré par un schéma de codage non scalable). Nous remarquons ainsi que le schéma proposé bien que possédant une telle scalablilité, ne présente qu'un léger surcoût par rapport au schéma non scalable.

TABLE 2. Exemple de débit(bpp)-PSNR(dB) du schéma de codage non scalable de Balloons vue 3 image 1

Q_p	Th_{Quad}	$Coût_{Grille_{Prof}}$	$Coût_{Prof}$	$PSNR_{Prof}$	$Coût_{Grille_{Tex}}$	$Coût_{Tex}$	$PSNR_{Tex}$
8	5	0.024	0.295	54.55	0.043	1.797	41.93
22	14	0.009	0.077	41.82	0.043	0.474	37.33
36	24	0.007	0.053	39.67	0.032	0.261	35.07
50	33	0.005	0.036	37.06	0.026	0.18	33.38
64	42	0.004	0.03	36.21	0.022	0.131	31.82
78	52	0.003	0.023	33.23	0.018	0.103	30.28
92	61	0.003	0.02	32.42	0.015	0.085	28.86
113	75	0.002	0.014	29.96	0.012	0.064	27.07
127	84	0.001	0.012	29.12	0.01	0.055	26.23

TABLE 3. Exemple de débit(bpp)-PSNR(dB) et de surcoût (bpp) du schéma scalable de Balloons vue 3 image 1

Q_p	Th_{Quad}	$Coût_{Tex_{BR}}$	$PSNR_{Tex_{BR}}$	$Coût_{\Delta_{Grille}}$	$Coût_{\Delta_{Tex}}$	$PSNR_{\Delta_{Tex}}$	**Surcoût**
8	5	0.637	24.94	0.041	1.428	41.89	**0.265**
22	14	0.118	21.21	0.04	0.433	37.25	**0.075**
36	24	0.072	20.53	0.03	0.243	34.93	**0.051**
50	33	0.046	19.69	0.024	0.173	33.17	**0.036**
64	42	0.036	19.45	0.02	0.126	31.54	**0.029**
78	52	0.028	18.82	0.017	0.101	29.91	**0.025**
92	61	0.025	18.7	0.014	0.083	28.42	**0.022**
113	75	0.019	18.21	0.011	0.064	26.59	**0.018**
127	84	0.017	18.18	0.009	0.055	25.8	**0.017**

TABLE 4. Exemple de débit(bpp)-PSNR(dB) du schéma de codage non scalable de Undodancer vue 1 image 250

Q_p	Th_{Quad}	$Coût_{Grille_{Prof}}$	$Coût_{Prof}$	$PSNR_{Prof}$	$Coût_{Grille_{Tex}}$	$Coût_{Tex}$	$PSNR_{Tex}$
8	5	0.006	0.046	47.7	0.034	2.452	42.32
22	14	0.004	0.029	44.13	0.04	1.001	36.57
36	24	0.003	0.023	42.17	0.039	0.566	33.63
50	33	0.002	0.018	38.68	0.037	0.375	31.83
64	42	0.002	0.014	36.23	0.033	0.259	30.43
78	52	0.001	0.011	34.21	0.029	0.192	29.37
92	61	0.001	0.01	33.9	0.026	0.146	28.49
113	75	0.001	0.007	31.51	0.022	0.101	27.2
127	84	0.001	0.006	30.23	0.019	0.082	26.39

TABLE 5. Exemple de débit(bpp)-PSNR(dB) et de surcoût (bpp) du schéma scalable de Undodancer vue 1 image 250

Q_p	Th_{Quad}	$Coût_{Tex_{BR}}$	$PSNR_{Tex_{BR}}$	$Coût_{\Delta_{Grille}}$	$Coût_{\Delta_{Tex}}$	$PSNR_{\Delta_{Tex}}$	**Surcoût**
8	5	0.149	20.17	0.034	2.28	42.38	**0.054**
22	14	0.065	18.29	0.04	0.93	36.66	**0.032**
36	24	0.042	17.95	0.039	0.54	33.72	**0.037**
50	33	0.028	17.64	0.036	0.36	31.94	**0.032**
64	42	0.019	17.26	0.032	0.26	30.56	**0.027**
78	52	0.016	16.8	0.028	0.19	29.51	**0.024**
92	61	0.014	16.78	0.026	0.15	28.63	**0.02**
113	75	0.01	16.65	0.021	0.10	27.38	**0.017**
127	84	0.009	16.62	0.018	0.08	26.55	**0.016**

4.6.4 *Étude de la complexité*

Nous avons étudié la complexité du schéma scalable proposé en terme de temps d'exécution. Le logiciel utilisé est le `Vtunes`, qui permet de mesurer le temps d'exécution de chaque fonction à part comme le montre la FIGURE 92.

TABLE 6. Moyenne des surcoût des différentes images de MPEG 3D.

Image	Balloons vue 5 image 1	Bookarrival vue 8 image 33	Gtfly vue 1 image 127	Kendo vue 1 image 1	Newspaper vue 6 image 1	Undodancer vue 5 image 250	**Moyenne**
Surcoût (bpp)	0.056	0.062	0.026	0.036	0.102	0.0277	**0.051**
Surcoût (%)	14.4%	17%	9.6%	13.3%	20 %	8.7%	**15** %

Ensuite, le code du schéma scalable est implémenté sur un `Raspberry pi` dont les caractéristiques sont données dans la FIGURE 93. La FIGURE 94 donne une idée sur le temps d'exécution du code sur le `Raspberry pi`. Le code n'étant pas optimisé, le temps d'exécution est élevé.

Pour plus de détails sur l'étude de la complexité, vous pouvez se référer au rapport de Master [84].

4.7 CONCLUSION

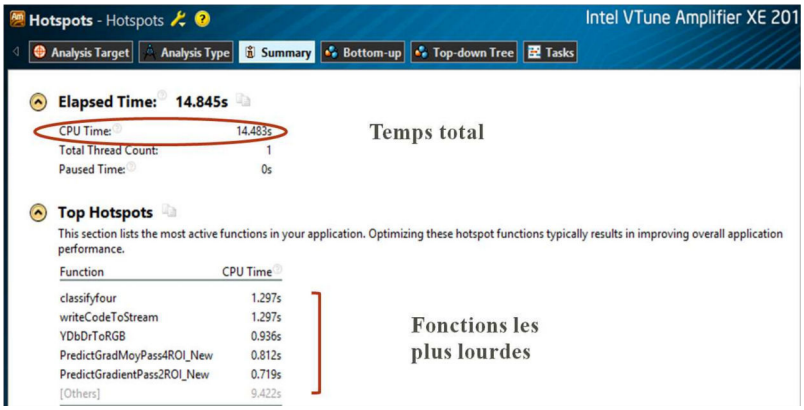
Dans ce chapitre, nous avons présenté un schéma de codage **joint** 2D+Z **scalable** basé sur le LAR. Dans un premier temps, nous avons introduit le codeur LAR, sur lequel nous avons implémenté le schéma scalable proposé.

Ensuite, nous avons détaillé le schéma de codage à basse résolution, où la texture est codée à basse résolution suivant la grille de profondeur et la profondeur est codée conjointement avec cette texture, avec une approche LARP (LAR pour Profondeur). Le LARP consiste en deux parties : une meilleure prédiction suivie d'un post-traitement adaptatif. D'une part, l'outil de "Meilleur Prédicteur" ne code pas la profondeur indépendamment de la texture, mais il exploite la forte corrélation entre la texture et la profondeur afin d'améliorer les performances de codage. Le meilleur prédicteur de la luminance de la texture, sélectionné a posteriori, est appliqué à la profondeur associée pour une meilleure prédiction. D'autre part, la carte reconstruite est soumise à un post-traitement : il s'agit d'une interpolation adaptative, éliminant d'une manière homogène les effets de blocs de la carte de profondeur.

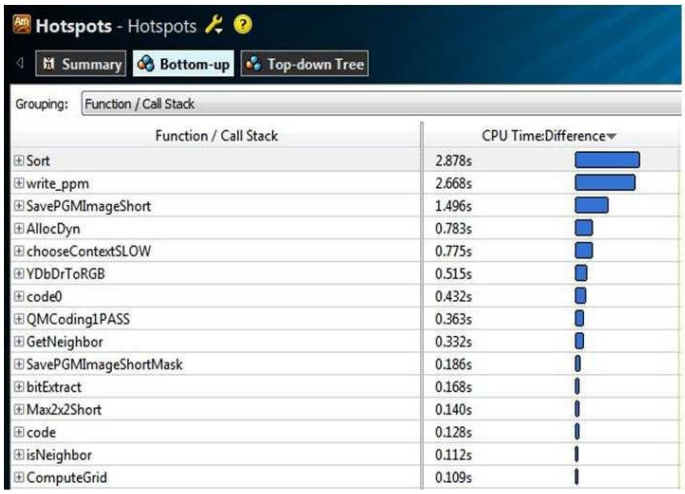
L'outil proposé est comparé à JPEG et JPEGXR pour la compression avec perte et à JPEGLS pour la compression sans perte. Suivant les résultats objectifs, l'outil proposé diminue le débit de la carte de profondeur par rapport au LAR classique, au JPEG et au JPEGXR. Malgré que le JPEGLS donne des résultats plus performants que l'outil proposé pour la compression sans perte, ce dernier implémenté sur le LAR, effectue efficacement un codage avec et sans perte. En outre, les résultats subjectifs montrent que l'outil proposé améliore la qualité visuelle des vues synthétisées, notamment sur les contours des objets.

Enfin, le schéma de raffinement de qualité est présenté. Il consiste en un raffinement de la grille simple de la profondeur suivie par le raffinement de la texture. Les expérimentations ont montré que le schéma proposé assure une scalabilité pour un léger surcoût par rapport au schéma de codage non scalable.

Un tel outil peut être utilisé par une multitude de codeurs prédictifs. Dans le chapitre suivant, nous explorons les aspects sémantiques des images 3D.



(a)



(b)

FIGURE 92. Exemple de résultats de `Vtunes`.

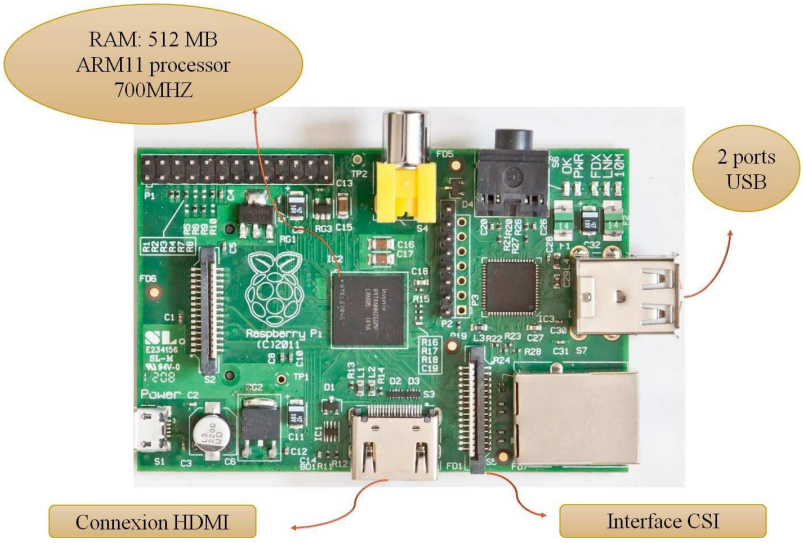


FIGURE 93. `Raspberry pi` utilisé pour l'implémentation du code du schéma scalable.

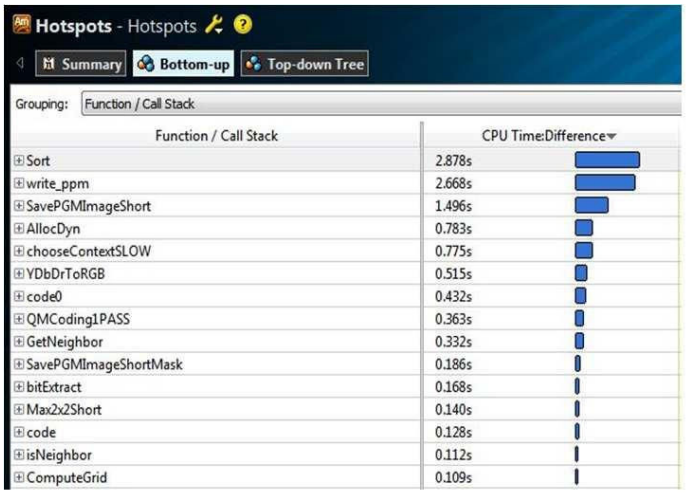


FIGURE 94. Résultats de temps d'exécution sur le `Raspberry pi` donné par `Vtunes`.

Deuxième partie

CODAGE PAR RÉGION D'INTÉRÊT 3D

ÉTAT DE L'ART

5 ÉTAT DE L'ART EN SEGMENTATION DES IMAGES 3D

Une image vaut mille mots.

— Confucius

Objectifs spécifiques du chapitre :

- **Connaître** et **Comprendre** les méthodes de segmentation 3D de l'État de l'Art.
- **Appliquer** et **Analyser** ces méthodes .

5.1 INTRODUCTION

Comme mentionné au début de cette mémoire, de nouvelles applications en image dans le domaine 3D voient le jour telles que la post-production, la reconnaissance d'objets, l'édition d'images, la vidéo-surveillance et l'indexation automatique. Ces applications exigent en premier lieu des fonctionnalités avancées pour le codage telles que la scalabilité en résolution et en qualité, la faible complexité, l'unicité de la compression avec et sans pertes, etc. Mais plus encore que pour le domaine 2D, ce type d'applications met en exergue le principe de composition de scènes à partir d'objets. Afin que cette caractéristique importante puisse ensuite être exploitée, il faut donc disposer initialement, d'un point de vue représentation, d'un niveau sémantique élevé. Un objet est toujours défini comme la composition de régions homogènes suivant différents critères de cohérence possibles liés par exemple au mouvement, à la connexité spatiale, ou encore appris par apprentissage. Dans cette étude, nous n'abordons pas les aspects représentation au niveau objets à proprement parlé, mais uniquement au niveau régions. Les techniques de séparation de l'image en régions sont couramment appelées segmentation. Elles sont utilisées depuis déjà longtemps dans le domaine 2D, et leurs principales limitations sont bien connues :

- problème d'initialisation des paramètres de segmentation [85],
- problème de similarité de couleur entre le premier plan et le fond [85],
- problème de changement d'illumination dans la scène [85],
- problème d'ombres dans la scène [85],
- problème de fonds entrelacés [85].

La représentation 3D de la scène, notamment à travers le format 2D+Z, permet de résoudre en partie certains problèmes, par le fait que la profondeur apporte une information importante et relativement facilement exploitable. Toutefois, la contrepartie est que l'introduction de cette nouvelle dimension augmente significativement la complexité de la segmentation de la scène. Les techniques de segmentation 2D+Z existantes sont pour la très grande majorité des extensions des techniques 2D. Parmi les approches existantes, les méthodes basées "graphe" sont les plus répandues. L'intérêt de telles méthodes est de fournir une représentation compacte, structurée, complète et facile à manipuler.

Nous introduisons ainsi dans la première partie de ce chapitre, un exemple classique de technique de segmentation 2D basée graphe (Section 5.2). Ensuite, dans la deuxième partie, nous présentons les différentes approches adoptées pour la segmentation 2D+Z (Section 5.3).

5.2 SEGMENTATION 2D BASÉE GRAPHE

Cette section introduit brièvement les algorithmes de segmentation 2D basés graphe. Ce type d'algorithmes suppose qu'une image peut être représentée par un graphe non orienté $G =< P, V >$, qui est défini comme un ensemble de nœuds (P) et un ensemble d'arêtes (V) qui lient ces nœuds. Les nœuds du graphe représentent les pixels de l'image et les arêtes du graphe représentent les liens entre les pixels adjacents. Chaque arête est affecté d'un poids $w(i, j)$

mesurant la ressemblance entre les deux nœuds/régions i et j qu'il relie (voir FIGURE 95). Ces poids sont calculés en utilisant la similarité de couleur ou d'intensité, le contraste, etc.... La segmentation de l'image consiste ensuite à fusionner les régions adjacentes suivant un critère de fusion. Les différents algorithmes vont ensuite essentiellement se distinguer par ce critère de fusion. Dans ce qui suit, nous présentons les critères les plus répandus et discutons leurs avantages et inconvénients.

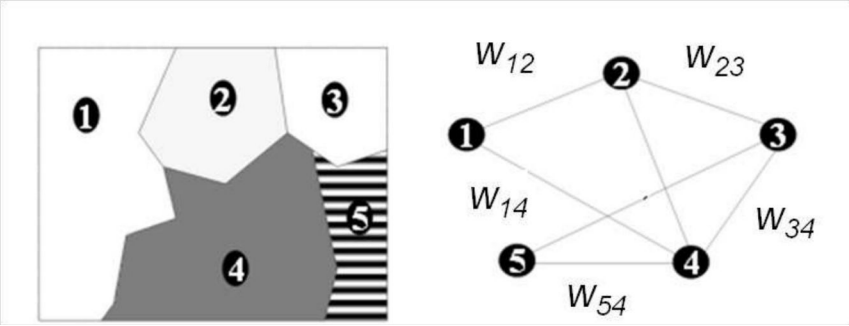


FIGURE 95. Segmentation en régions et graphe d'adjacence associé [87].

5.2.1 *Critère de fusion par optimisation d'énergie*

Un exemple typique d'algorithme de segmentation utilisant le critère de fusion par optimisation d'énergie est utilisé dans [86]. La valeur attribuée à chaque arête est dite "énergie".

Soit $L = (L_1, ..., L_p, ...L_P)$ un vecteur d'étiquettes dont l'élément L_p indique l'étiquette du pixel p. Pour une segmentation en deux régions par exemple, chaque pixel peut appartenir soit au fond ($L_p = 0$) soit au premier plan ($L_p = 1$).

Une fonction d'énergie $E(L)$ est définie par l'expression (27).

$$E(L) = \sum_{p \in P} E_p(L_p) + \lambda \sum_{(p,q) \in V} E_(p,q)(L_p, L_q) \tag{27}$$

avec :

- $E_p(L_p)$ correspond à l'énergie basée région, qui représente le coût quand l'étiquette du pixel p est L_p,
- $E_{(p,q)}(L_p, L_q)$ correspond à l'énergie basée périphérie, qui représente le coût quand les étiquettes des pixels adjacents p et q, sont L_p et L_q respectivement,
- $\lambda \in [0, 1]$ indique l'importance de l'énergie basée région contre celle basée périphérie.

Enfin, l'algorithme d'optimisation max-flow [87] est utilisé pour minimiser la fonction d'énergie. Le vecteur L optimisant la fonction d'énergie $E(L)$ du graphe définit ainsi la segmentation optimale. À l'issue de cet algorithme, les nœuds sont groupés sous différentes classes, ce qui est équivalent à faire appartenir chaque pixel à une région correspondante. Les régions sont définies comme des parties connexes d'une image possédant une propriété commune. L'ensemble de telles régions constituent ainsi des objets représentant une entité sémantique cohérente dans l'image.

Les algorithmes utilisant le critère de fusion par optimisation de l'énergie du graphe permettent d'ajuster la finesse de la simplification où la signification de la ressemblance entre deux régions peut ainsi être adaptée au niveau désiré. Toutefois, ces algorithmes possèdent plusieurs inconvénients :

- une grande complexité par rapport aux autres algorithmes,
- un temps de simplification d'un graphe dépendant de la complexité du graphe initial.

5.2.2 *Critère de fusion par seuillage*

Un exemple typique d'algorithme de segmentation utilisant le critère de fusion par seuillage est utilisé dans [88]. Pour chaque arête, une distance est calculée à partir de la valeur attribuée à l'arête et des grandeurs géométriques, tels que la surface de chaque région et le périmètre commun aux deux régions. Cette distance traduit ainsi le coût de fusion $\mathrm{Coût}(R_A, R_B)$ des deux régions adjacentes R_A et R_B. Chaque distance attribuée à une arête est comparée ensuite à un seuil, qui permet alors de décider la fusion ou non des deux régions R_A et R_B (voir Eq. 28). Le mécanisme de simplification est appliqué itérativement sur le graphe jusqu'à ce qu'aucune fusion ne soit plus possible.

$$\begin{aligned} \mathrm{Coût}(R_A, R_B) \quad &= \quad f(w(R_A, R_B), \mathrm{surf}(R_A), \mathrm{surf}(R_B), \mathrm{perim}(R_A,\ R_B)) \\ &\quad\ \ \textbf{Si}\ \mathrm{Coût}(R_A, R_B) < \mathrm{Seuil}\ , \text{fusionner}\ R_A \text{ et } R_B \end{aligned} \tag{28}$$

Les algorithmes utilisant le critère de fusion par seuillage possèdent plusieurs avantages par rapport aux autres algorithmes :

- une faible complexité,
- un ajustement du degré de granularité de la segmentation par un simple changement du seuil,
- une possibilité de pondération du poids d'une arête par la surface de la région, ce qui permet d'éviter l'apparition des petites régions en les fusionnant avec les autres régions adjacentes.

Toutefois, étant un algorithme itératif, le temps de segmentation dépend fortement de la taille de l'image et de sa complexité.

5.2.3 *Critère de fusion par "coupe minimale normalisée" (Normalized Graph Cut)*

Un exemple typique d'algorithme de segmentation utilisant le critère de fusion par coupe minimale normalisée est utilisé dans [89]. Le principe de telles méthodes est de partitionner le graphe d'adjacence correspondant en réalisant des coupures récursives minimisant un coût donné. Ce principe peut être résumé par les 2 étapes suivantes :

1- Séparer le graphe G en deux ensembles disjoints A et B, tels que $A \cup B = G$ et $A \cap B = \emptyset$, en éliminant simplement les arêtes liant les deux ensembles. La bipartition optimale du graphe est celle qui le sépare en deux régions les moins ressemblantes. En d'autres termes, la bipartition optimale est celle qui minimise la valeur de disassociation Ncut, qui est fonction des poids des arêtes liant les deux ensembles à séparer. Il traduit ainsi la ressemblance entre ces deux ensembles (voir Eq. 29).

$$\mathrm{Ncut} = \frac{\mathrm{cut}(A, B)}{\mathrm{assoc}(A, V)} + \frac{\mathrm{cut}(A, B)}{\mathrm{assoc}(B, V)} \tag{29}$$

avec $\mathrm{assoc}(A, V) = \sum_{u \in A, t \in V} w(u, t)$ est la somme des poids des arêtes liant l'ensemble A avec tous les nœuds du graphe, $\mathrm{assoc}(B, V)$ est définie de manière similaire pour l'ensemble B, et $\mathrm{cut}(A, B)$ est la somme des poids des arêtes liant les ensembles A et B.

2- Décider si la partition courante doit être sub-divisée et partitionner les parties séparées de manière récursive si nécessaire. La partition d'un ensemble s'arrête si Ncut dépasse une certaine limite.

Les algorithmes utilisant le critère de fusion par coupe minimale normalisée possèdent généralement une grande complexité dans le calcul de la valeur de Ncut.

L'information de profondeur des objets dans la scène constitue une information sémantique importante. Elle a ainsi été utilisée suivant plusieurs approches afin d'améliorer la performance de la segmentation. Ces approches peuvent être groupées en deux catégories : 1) les approches utilisant la profondeur pour raffiner (simplifier) a posteriori la segmentation fine basée couleur uniquement, 2) celles utilisant la profondeur en parallèle avec la texture pour segmenter la scène.

5.3.1 *Approches utilisant la profondeur pour simplifier a posteriori la segmentation*

Un exemple de méthodes est donné dans [90]. Il s'agit dans un premier temps de segmenter la scène à partir de la texture uniquement. L'hétérogénéité de la texture produit alors une segmentation fine et détaillée de la scène. La profondeur est ensuite utilisée pour fusionner certaines régions et simplifier la carte sur-segmentée.

Le principe de la méthode de *Cigla et al.* comporte donc deux grandes étapes :

1- segmentation basée couleur uniquement : la texture est segmentée par un algorithme de segmentation 2D basé graphe classique, tel que ceux présentés précédemment. L'image est représentée dans un premier temps par un graphe, dont les nœuds correspondent aux pixels dans l'image de texture, et les liens représentent la similarité de couleur entre les pixels. Chaque lien est caractérisé par un poids. Le poids d'un lien $\in$ à l'intervalle $[0, 1]$, avec le poids égal à 1 est attribué au lien entre deux pixels de même couleur. Après la construction du graphe, la segmentation se fait par l'élimination récursive des liens à poids faible, suivant un critère de mesure prédéfini. À l'issue de cette étape, l'image sera représentée par des régions connectées sur-segmentées.

2- simplification de la sur-segmentation : les régions obtenues par la segmentation basée couleur, sont ensuite reliées par des liens pondérés. Le poids $w_{i,j}$ du lien connectant deux régions R_i et R_j de la carte sur-segmentée, est calculé suivant l'équation (30).

$$w_{i,j} = \begin{cases} e^{-D_{ij}^2/\sigma_d}.e^{-I_{ij}^2/\sigma_i} & \text{si} \quad |SL_i - SL_j|^2 < R \\ 0 & \text{si} \quad \textit{autre} \end{cases} \tag{30}$$

σ_d et σ_i correspondent aux facteurs de pondération de la similarité de la profondeur et de la texture. Ensuite, D_{ij}^2 et I_{ij}^2 sont calculés par les équations (31) et (32), respectivement.

$$D_{ij}^2 = \frac{1}{N_B} \sum_{x \in B_i,\ y \in B_j} |D(x) - D(y)|^2 \tag{31}$$

$$I_{ij}^2 = \frac{1}{(255)^2 N_B} \sum_{x \in B_i,\ y \in B_j} |I(x) - I(y)|^2 \tag{32}$$

(33)

Dans les équations (31) et (32), D et I représentent respectivement la profondeur et la texture. B_i indique l'ensemble des pixels de périphérie de la région R_i. Finalement, SL_i indique la position moyenne de la région R_i.

Les facteurs σ_d et σ_i déterminent l'importance de l'information correspondante. De cette manière, plusieurs possibilités de simplification existent. Les auteurs dans [90] choisissent les facteurs de pondération de manière égale, de sorte que les informations issues de la texture et de la profondeur soient de la même importance.

De telles approches utilisant la profondeur pour raffiner a posteriori une carte de segmentation détaillée, possèdent deux limitations majeures :

- Dépendance de la segmentation initiale : la sur-segmentation initiale affecte la précision de la segmentation finale de l'image. Les erreurs initiales se propagent ainsi vers la segmentation finale.

- Problème de similitude de couleur entre le fond et le premier plan non complètement résolu : deux objets de même couleur mais appartenant à deux profondeurs différentes, seront fusionnés lors de la première étape de manière définitive. La profondeur n'est pas donc bien exploitée pour séparer le fond du premier plan.

5.3.2 *Approches utilisant la profondeur en parallèle avec la texture pour la segmentation*

Un premier exemple de méthodes utilisant la profondeur en parallèle avec la texture pour la segmentation est donné dans [91]. La méthode de segmentation proposée par *Dai et al.* consiste à localiser la région d'objet saillant par un algorithme de détection de saillance, et à calculer les énergies du graphe de segmentation en fonction de la texture uniquement, suivi d'un ajustement des énergies du graphe initial, en fonction de la profondeur avant de débuter la segmentation. La méthode utilisée comprend 2 grandes étapes :

1- Détection de la région d'objet saillant : un seuil de profondeur est tout d'abord choisi. Les pixels dont la valeur de la profondeur est supérieure à ce seuil sont ainsi considérés comme des candidats des objets du premier plan. Inversement, les pixels dont la valeur de la profondeur est inférieure au seuil sont considérés comme des candidats du fond. Un filtre gaussien est ensuite appliqué aux pixels candidats des objets du premier plan, afin de limiter leur nombre. La valeur de visibilité (*conspicuity*) est alors calculée par la méthode de "*Center-Surround*", pour chacun des pixels candidats des objets du premier plan. Une région contenant des pixels dont la valeur de visibilité est supérieure à un certain seuil, est ainsi considérée comme une région d'objet saillant. Cette étape peut être résumée par la FIGURE 96. Un exemple de détection d'objet saillant est illustré dans la FIGURE 97.

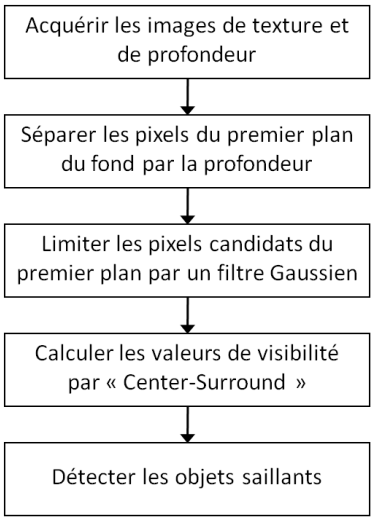


FIGURE 96. Algorithme de détection des objets saillants.

(a) texture (b) profondeur (c) pixels candidatds (d) carte de saillance (e) region de saillance

FIGURE 97. Exemple de détection des objets saillants.

TABLE 7. Ajustement de la fonction d'énergie basée région

Texture, Profondeur	Décision de consistance	Ajustement de l'énergie basée région
$E_p(O_p) > E_p(B_p)$, prof > seuil	accord	incrémenter $E_p(O_p)$
$E_p(O_p) < E_p(B_p)$, prof < seuil		incrémenter $E_p(B_p)$
$E_p(O_p) > E_p(B_p)$, prof < seuil	contradiction	atténuer $E_p(O_p)$
$E_p(O_p) < E_p(B_p)$, prof > seuil		atténuer $E_p(B_p)$

2- Segmentation basée graphe avec ajustement des termes d'énergie (voir FIGURE 98) : en se basant sur le résultat de détection de la région d'objet saillant, une configuration initiale est faite : les pixels n'appartenant pas à la région de saillance détectée sont considérés appartenir au fond, alors que les pixels appartenant à cette région sont considérés comme un ensemble de pixels "incertains". Les énergies du graphe $E_p(L_p)$ et $E_{(p,q)}(L_p, L_q)$ sont ensuite calculées en se basant sur l'analyse de couleur uniquement en utilisant le *Gaussian Mixture Model (GMM)*. Le terme d'énergie $E_p(L_p)$ est divisé en $E_p(O_p)$ et $E_p(B_p)$, où $E_p(O_p)$ décrit le degré de similarité, en terme de couleur, entre le pixel et l'objet saillant, et $E_p(B_p)$ décrit le degré de similarité en termes de couleur, entre le pixel et le fond.

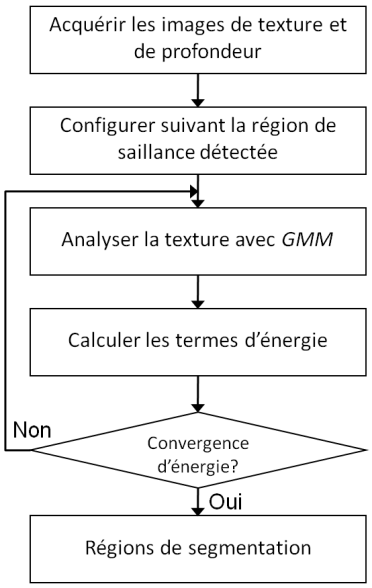


FIGURE 98. Algorithme de segmentation avec ajustement des termes d'énergie.

Enfin, un ajustement des termes d'énergie est appliqué, suivant une décision de consistance entre la texture et la profondeur. L'ajustement de l'énergie basée région ($E_p(L_p)$) est réalisé comme le montre le tableau 7. Si l'information de couleur montre que le pixel est similaire à l'objet saillant, et la profondeur montre encore que ce pixel est localisé dans une région correspondante au premier plan de l'image, alors la couleur et la profondeur sont en accord. Dans ce cas là, l'énergie $E_p(O_p)$ est alors incrémentée, afin d'augmenter la possibilité du pixel d'appartenir à l'objet saillant. De même, si l'information de couleur montre que le pixel est similaire au fond, et la profondeur montre encore que ce pixel est localisé dans le fond de l'image, alors la couleur et la profondeur sont encore en accord, et l'énergie $E_p(B_p)$ est dans ce cas là incrémentée, afin d'augmenter la possibilité du pixel d'appartenir au fond. Par contre, si l'information de couleur montre que le pixel est similaire à l'objet saillant, alors que la profondeur montre qu'il est localisé au fond, ou inversement, une incohérence existe donc entre la couleur et la profondeur. Les énergies $E_p(O_p)$ et $E_p(B_p)$ sont diminuées, afin d'atténuer la possibilité du pixel d'appartenir à l'objet saillant et au fond.

TABLE 8. Ajustement de la fonction d'énergie basée périphérie

Texture, Profondeur	Décision de consistance	Ajustement de l'énergie basée périphérie
couleur similaire, profondeur similaire	accord	aucun changement
couleur différente, profondeur différente		
couleur similaire, profondeur différente	contradiction	Atténuer la valeur de $E_p(B_p)$
couleur différente, profondeur similaire		

D'autre part, l'ajustement de l'énergie basée périphérie ($E_{(p,q)}(L_p, L_q)$) est réalisé comme le montre le tableau 8. Une telle énergie dépend uniquement des relations entre les pixels adjacents, et son ajustement se base donc sur la vérification de la similarité de couleur et de profondeur entre les pixels adjacents. Si les pixels adjacents ont une couleur ou profondeur similaires, alors les deux sont en cohérence. L'énergie basée périphérie conserve donc la même valeur calculée à partir de la couleur. Par contre, si seule la couleur est similaire, alors il y a une incohérence entre la couleur et la profondeur, et la valeur d'énergie basée périphérie est donc atténuée. Le processus est répété d'une manière itérative jusqu'à ce que la segmentation devienne stable.

Un autre exemple d'approches de segmentation 2D+Z appartenant à la seconde catégorie où la profondeur est directement introduite dans le calcul de l'énergie du graphe, est donné dans [92].

- Dans un premier temps, certains pixels de la texture et les pixels correspondants dans la profondeur, sont étiquetés manuellement comme appartenant au fond ou au premier plan. Les centres de ces deux régions sont ainsi calculés par la méthode de K-means.
- Ensuite, en contraste avec l'équation (27) du calcul de la fonction d'énergie du graphe en fonction de la couleur uniquement, les auteurs dans [92] utilisent la couleur et la profondeur dans le calcul de la fonction d'énergie du graphe, suivant l'équation (34).

$$E(L) = \left[\theta \sum_{p\in P} E^c_p(L_p) + (1-\theta) \sum_{p\in P} E^d_{p\in P}(L_p)\right] + \lambda \left[\theta \sum_{(p,q)\in V} E^c_{(p,q)}(L_p, L_q) + (1-\theta) \sum_{(p,q)\in V} E^d_{(p,q)}(L_p, L_q)\right] \quad (34)$$

avec θ correspondant au poids d'importance entre le terme de couleur et celui de la profondeur, les exposants c, d des termes d'énergies correspondant aux termes de couleur et de profondeur, respectivement. Les autres variables ont les mêmes significations que l'équation (27).

Pour calculer l'énergie basée région, *He et al.* utilisent la distance entre chaque pixel non étiqueté et le centre du fond d'une part et le centre du premier plan d'autre part. Par contre, pour le calcul de l'énergie basée périphérie, ils utilisent le gradient ou le contraste entre les pixels adjacents.

Le paramètre θ varie de 0 à 1. Pour $\theta = 0$, la segmentation dépend uniquement de la profondeur, alors que pour $\theta = 1$, la segmentation ne dépend que de la texture. La texture et la profondeur contribuent dans la segmentation lorsque $\theta \in [0, 1]$. Ainsi, les cartes de profondeur contenant de contrastes élevés, compensent le désavantage de la segmentation basée couleur uniquement. En revanche, pour les images dont les objets appartiennent à des plans très proches, la texture toute seule peut être plus pertinente.

5.4 CONCLUSION

Ce chapitre a introduit différentes méthodes de segmentation 2D+Z de la littérature. Ces méthodes sont généralement des extensions des méthodes de segmentations 2D basées graphe. Les approches adoptées peuvent être groupées en deux catégories. La première catégorie consiste à raffiner a posteriori le résultat de segmentation 2D basée couleur uniquement. Des régions de la carte de segmentation 2D sont fusionnées si une contrainte basée profondeur est respectée. La deuxième catégorie d'approches de segmentation 2D+Z consiste à utiliser la profondeur en parallèle avec la texture pour la segmentation de la scène. La fonction d'énergie du graphe est modifiée de sorte qu'elle sera fonction de la couleur et de la profondeur.

Devant cette diversité d'algorithmes de segmentation 2D+Z, l'algorithme de segmentation basé graphe utilisant le critère de fusion par seuillage présente plus d'avantages par rapport à d'autres algorithmes. L'objectif du chapitre suivant, n'est pas ainsi de proposer simplement un nouvel algorithme de segmentation 2D+Z, mais plus globalement un schéma global de représentation et codage niveau région dédié aux images 3D. Nous introduisons dans la première partie du chapitre suivant un schéma d'"Autofocus 3D simple" qui consiste à se focaliser sur une zone de profondeur d'intérêt dans la scène. Ensuite, dans la deuxième partie du chapitre, nous proposons schéma d'"Autofocus 3D avancé" qui s'appuie en supplément sur une segmentation sémantique de la scène 3D. Ces deux schémas vont considérer globalement les problèmes de compression et de représentation, notamment les problèmes de distorsions au niveau des contours, afin de préserver une cohérence entre la texture, la profondeur et les régions et assurer une haute qualité de synthèse de vue.

CONTRIBUTIONS

6 AUTOFOCUS 3D ET SEGMENTATION SÉMANTIQUE

Avoir une représentation 3D relief
conduit à repenser les organes étudiés
et parfois proposer de nouveaux
mécanismes ou de nouvelles approches
qui pourront être alors testés biologiquement.

— Aassif Benassarou *et al.*, Visualisation 3D relief du vivant [20]

Objectifs spécifiques du chapitre :

- **Synthétiser** deux schémas joints de représentation fine et de codage basé contenu d'images.
- **Évaluer** les deux schémas proposés.

6.1 INTRODUCTION

Les premières générations de codage d'images et de vidéo étaient indépendantes du contenu de ces images. Toutefois, elles offraient une bonne efficacité de codage au regard de leur complexité. Les nouveaux standards H.264 et HEVC, en introduisant une adaptation partielle au contenu à travers des blocs à taille variable, ont permis d'augmenter significativement les performances de codage. Les techniques de codage dites de seconde génération, proposées initialement par *Kunt* [93, 94] avaient pour objectif de mieux prendre en compte le contenu sémantique des images. Ces approches s'appuient essentiellement sur la représentation des données visuelles en terme de régions, définies par leurs contours et leurs textures, et correspondant potentiellement à des objets ou parties d'objets dans l'image. Le grand intérêt des approches basées région est qu'*elles tendent à combler le fossé séparant les systèmes numériques des systèmes visuels humains (SVH) pour le traitement d'une image et sa perception [78].*

Malheureusement, les représentations à haut niveau sémantique sont en général antinomiques avec une efficacité de compression. Ainsi les algorithmes de représentation en régions présentés précédemment permettent d'extraire uniquement les formes des objets de la scène indépendamment du codage de leur contenu. La question alors est de savoir comment coupler une représentation des contours qui peut être de résolution "pixelique" avec un schéma de codage qui peut être lui basé bloc.

Nous proposons ainsi dans ce chapitre deux solutions tentant d'unifier les notions de forme et de contenu, basées sur le schéma scalable 2D+Z proposé dans le Chapitre 4, et exploitant la profondeur comme information sémantique additionnelle :

1) Dans la première partie du chapitre, nous proposons un schéma joint de représentation et codage basé région de "Profondeur d'Intérêt (*Depth of Interest* DoI), appelé "Autofocus 3D simple". Le schéma de représentation de la profondeur d'intérêt DoI consiste en une extraction fine des objets situés à l'intérieur de la DoI. Le schéma de codage DoI consiste ensuite, lors du codage scalable 2D+Z, à assurer une haute qualité des objets situés à l'intérieur de la DoI, au détriment des autres zones. Le raffinement local de qualité apporte un rehaussement de qualité SNR et/ou de résolution. Dans la Section 6.2, nous détaillons le schéma d'Autofocus simple pour les images 3D et nous évaluons les résultats de représentation et de codage obtenus dans la Section 6.2.6.

2) Dans la deuxième partie du chapitre, nous proposons un schéma d'"Autofocus 3D avancé" basé sur une segmentation sémantique de la scène. La représentation sémantique en régions de la scène, basée sur la version basse résolution de l'image 3D, permet d'ajuster la segmentation de l'image 3D suivant un compromis entre la texture et la profondeur. Une technique d'Autofocus 3D analogue à la précédente permet ensuite de sélectionner un nombre de régions considérées comme Région d'Intérêt, afin de raffiner leur qualité à plus

haut débit. Dans la Section 6.3, nous détaillons le schéma d'Autofocus basé segmentation et nous évaluons ses performances dans la Section 6.3.4.

6.2 AUTOFOCUS 3D SIMPLE

Comme introduit précédemment, les cartes de profondeur, considérées comme une information sémantique additionnelle de la scène, peuvent être exploitées dans le contexte de codage par Région d'Intérêt (RoI). Le schéma d'Autofocus 3D proposé permet ainsi de se focaliser, lors du codage scalable 2D+Z, sur une zone de profondeur dans la scène, considérée comme une Profondeur d'Intérêt (DoI) (voir FIGURE 99). Nous présentons ainsi dans ce qui suit le principe d'"Autofocus 3D simple" (Section 6.2.1). Dans la Section 6.2.2, nous présentons ensuite le schéma global d'"Autofocus 3D simple". Enfin, la Section 6.2.6 fournit les résultats de représentation, de codage et de synthèse de vue du schéma d'Autofocus 3D simple proposé.

(a) (b) (c)

FIGURE 99. Focaliser sur la zone de profondeur d'intérêt : (a) Texture ; (b) représentation fine de objets ; (c) texture codée avec focalisation sur la DoI.

6.2.1 *Propriétés du schéma d'Autofocus 3D*

Du point de vue de la représentation et du codage, trois critères importants doivent être considérés pour les systèmes de codage par région d'intérêt : 1) la résolution spatiale de la RoI, 2) le coût du masque de la RoI, et 3) le choix de la qualité entre la RoI et la non-RoI. La plupart des codeurs de l'État de l'Art utilisent l'approche de codage RoI basé bloc, où l'unité de codage est un bloc de minimum 16x16 pixels, tels que JPEGXR, H.264 et HEVC. Si la résolution spatiale de la RoI est basée bloc, le coût du masque RoI est réduit, mais en contre-partie la qualité des contours de la RoI sera fortement dégradée. D'autre part, d'autres codeurs de l'État de l'Art tels que JPEG2000, utilisent l'approche de codage RoI basé pixels, mais imposent la différence de qualité entre la RoI et la non-RoI.

Contrairement aux codeurs de l'État de l'Art, l'Autofocus 3D simple proposé, s'appuyant sur le LAR, correspond mieux aux trois critères de codage RoI : 1) il s'appuie sur une résolution basée pixel, 2) le coût de transmission du masque est nul, seulement les deux valeurs limitant la profondeur d'intérêt sont codées, et 3) il permet de choisir n'importe quel niveau de qualité aussi bien pour la RoI que pour le fond. Le tableau 9 établit une comparaison fonctionnelle entre ces différents codeurs.

TABLE 9. Comparaison fonctionnelle de codage RoI

Codeur	Unité de Résolution de la RoI	Différentes Qualités pour Non-RoI/RoI	Transmission du Masque RoI
JPEG2K	pixel	Non	Non
JPEGXR	bloc	Oui (sauf sans perte)	Oui
H.264	bloc	Oui (sauf sans perte)	Oui
HEVC	bloc	Oui (sauf sans perte)	Oui
LAR	pixel	Oui	Non

6.2.2 *Schéma global d'Autofocus 3D simple*

Le schéma d'Autofocus 3D simple proposé est intégré dans le schéma de codage scalable 2D+Z présenté dans le Chapitre 4. Pour les applications 2D+Z, il peut être considéré que l'objet d'intérêt soit situé dans une zone de profondeur spécifique. Cet intervalle de profondeur, considéré comme paramètre d'entrée de notre schéma global d'Autofocus, définit la profondeur d'intérêt (*Depth of Interest* DoI). Dans un premier temps, le schéma de représentation de la DoI consiste à définir, de la carte de profondeur reconstruite, un masque binaire (Masque-DoI) couvrant la zone de profondeur d'intérêt. Ensuite, le schéma de codage DoI consiste à assurer une haute qualité pour la DoI au détriment du reste des zones de profondeur. Une haute qualité est assurée pour chacune des images de profondeur et de texture (voir FIGURE 100).

À cette fin, dans un premier temps un raffinement de la DoI est appliqué sur la carte de profondeur (Section 6.2.3) : la grille de QuadTree de la profondeur est adaptée pour avoir une plus haute résolution à l'intérieur de la DoI, et avoir ainsi des contours bien définis tout au long des objets d'intérêt. Un codage de la DoI est ensuite appliqué sur la texture (Section 6.2.5) : le schéma de codage du LAR basé RoI utilisant le masque binaire basé profondeur permet de choisir différentes qualités à l'intérieur et à l'extérieur de la DoI.

Les avantages du schéma d'Autofocus proposé sont : 1) il est basé contenu d'images, 2) le schéma de raffinement DoI est inséré comme un pré-traitement, nous évitons donc de modifier le codeur ce qui réduit sa complexité, 3) le Masque-DoI n'est pas transmis au décodeur, il n'y a pas donc de surcoût.

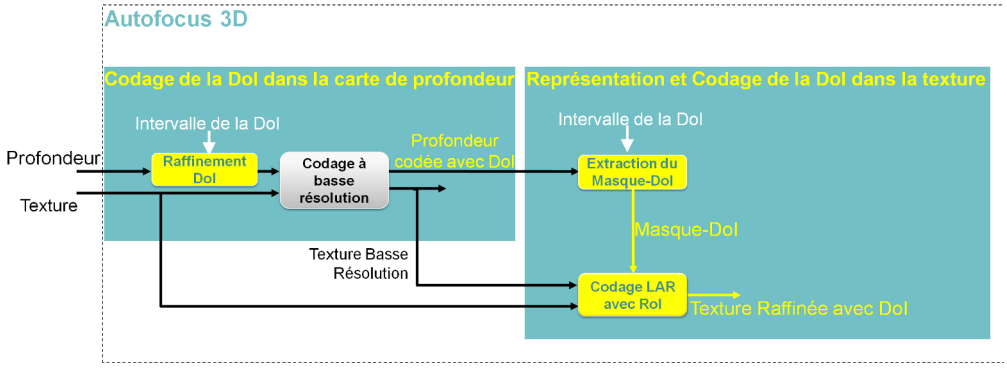


FIGURE 100. Schéma global du schéma d'Autofocus proposé.

6.2.3 *Codage de la DoI dans la carte de profondeur*

Comme mentionné précédemment, la qualité de la carte de profondeur codée est fortement liée aux distorsions introduites sur les contours. Un rehaussement de la qualité locale de la DoI

dans la carte de profondeur vise donc à raffiner la résolution spatiale. Le seuil du QuadTree du LAR étant fixe, une solution possible consiste à concevoir un nouveau schéma QuadTree avec un seuil adaptatif. Toutefois, nous avons recours à une autre solution conservant le schéma QuadTree original, mais modifiant la carte de profondeur initiale. Plus spécifiquement, il s'agit d'un ajustement de la dynamique de la carte de profondeur originale, à ce que la DoI ait le plus large intervalle dynamique, au détriment des autres zones de profondeur. Nous introduisons les notations suivantes :

- $\mathtt{In\_Prof}$: carte de profondeur originale d'entrée.
- $\mathtt{Out\_Prof}$: carte de profondeur reconstruite après ajustement de la dynamique.
- $Z_{\mathtt{in\_l}}$ et $Z_{\mathtt{in\_h}}$: limites inférieure et supérieure de l'intervalle de la DoI.
- $W_{\mathtt{in}}$: fenêtre d'intérêt d'entrée, avec $W_{\mathtt{in}} = Z_{\mathtt{in\_h}} - Z_{\mathtt{in\_l}}$.
- $Z_{\mathtt{out\_l}}$ et $Z_{\mathtt{out\_h}}$: limites inférieure et supérieure de l'intervalle ajusté de la DoI.
- $W_{\mathtt{out}}$: fenêtre de profondeur ajustée, avec $W_{\mathtt{out}} = \mathtt{F} \times W_{\mathtt{in}} = Z_{\mathtt{out\_h}} - Z_{\mathtt{out\_l}}$.

L'intervalle de profondeur $[Z_{\mathtt{in\_l}}, Z_{\mathtt{in\_h}}]$ et le coefficient $\mathtt{F}$ peuvent respectivement être considérés comme une approximation de la distance focale et du *F-number* (ou ouverture relative) du système optique. En particulier, le paramètre $\mathtt{F}$ contrôle le degré de finesse la DoI. Les paramètres $Z_{\mathtt{in\_l}}$ et $Z_{\mathtt{in\_h}}$ peuvent être ensuite sélectionnés soit par l'utilisateur soit d'une manière automatique par des techniques avancées d'extraction d'objets 3D telles que celles proposées dans [95] ou [96].

La fonction d'ajustement de la dynamique proposée est illustrée par la FIGURE 101, et l'algorithme correspondant est donné par l'algorithme 6.2.1. Nous posons la contrainte que la valeur du point milieu de l'intervalle de profondeur d'entrée soit inchangée.

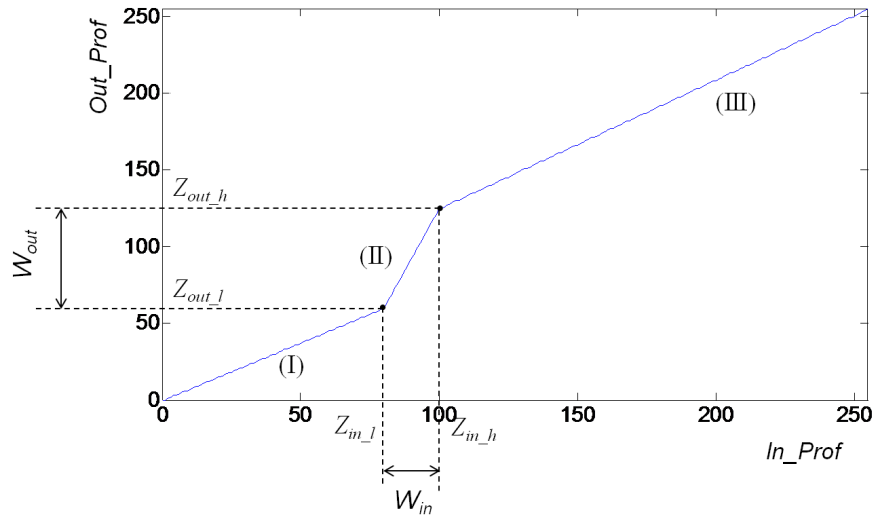


FIGURE 101. Ajustement de la dynamique des valeurs de la profondeur reconstruite.

La carte de sortie $\mathtt{Out\_Prof}$ est ensuite utilisée par le QuadTree pour calculer la grille de la profondeur, et la carte de profondeur originale est codée avec le LARP avec le même facteur de quantification appliqué sur toute la carte.

Algorithme 6.2.1 *: Ajustement de la Dynamique de la Carte de Profondeur Originale*

Entrée : In_Prof, Z_{in_l}, Z_{in_h}, F
Sortie : Out_Prof

$\delta W = W_{out} - W_{in} = (F-1) \times W_{in}$
$[Z_{out\_l},\ Z_{out\_h}]$ = **calcul_paramètres** $(Z_{in\_l},\ Z_{in\_h},\ \delta W)$
pour tout i **faire**
si $In\_Prof(i) \leqslant Z_{in\_l}$ **alors**

$$b = \frac{Z_{out\_l}}{Z_{in\_l}}$$

$Out\_Prof(i) = b \times In\_Prof(i).$ (I)
fin si
si $Z_{in\_l} < In\_Prof(i) < Z_{in\_h}$ **alors**

$$b = \frac{Z_{out\_h} - Z_{out\_l}}{Z_{in\_h} - Z_{in\_l}}$$

$c = Z_{out\_l} - b \times Z_{in\_l}$
$Out\_Prof(i) = b \times In\_Prof(i) + c.$ (II)
si $In\_Prof(i) > Z_{in\_h}$ **alors**

$$b = \frac{Z_{out\_h} - 255}{Z_{in\_h} - 255}$$

$c = 255 \times (1 - b)$
$Out\_Prof(i) = b \times In\_Prof(i) + c.$ (III)
fin si
fin pour
{retourner la carte de profondeur ajustée}
retourner Out_Prof

calcul_paramètres $(Z_{in\_l},\ Z_{in\_h},\ \delta W)$
Entrée : $Z_{in\_l}, Z_{in\_h}, \delta W$
Sortie : $Z_{out\_l},\ Z_{out\_h}$
$\delta W = W_{out} - W_{in} = (F-1) \times W_{in}$
si $Z_{in\_l} = 0$ **alors**
$Z_{out\_l} = 0$
$Z_{out\_h} = Z_{in\_h} + \delta W$
fin si
si $Z_{in\_h} = 255$ **alors**
$Z_{out\_l} = Z_{in\_l} - \delta W$
$Z_{out\_h} = 255$
fin si
autre

$$Z_{out\_l} = Z_{in\_l} - \frac{\delta W}{2}$$

$$Z_{out\_h} = Z_{in\_h} + \frac{\delta W}{2}$$

retourner Z_{out_l} et Z_{out_h}

6.2.4 *Résultats du QuadTree après le pré-traitement des cartes de profondeur*

Les tests sont réalisés sur les images 3D de référence fournies par MPEG. Nous appliquons le raffinement de la DoI sur les cartes de profondeur correspondantes de Balloons et UndoDancer. Les paramètres à contrôler sont :

- Qp, Th_{Quad} pour la qualité de l'image 2D+Z compressée
- Z_{in_l}, Z_{in_h} et F pour l'ajustement de la dynamique de la DoI.

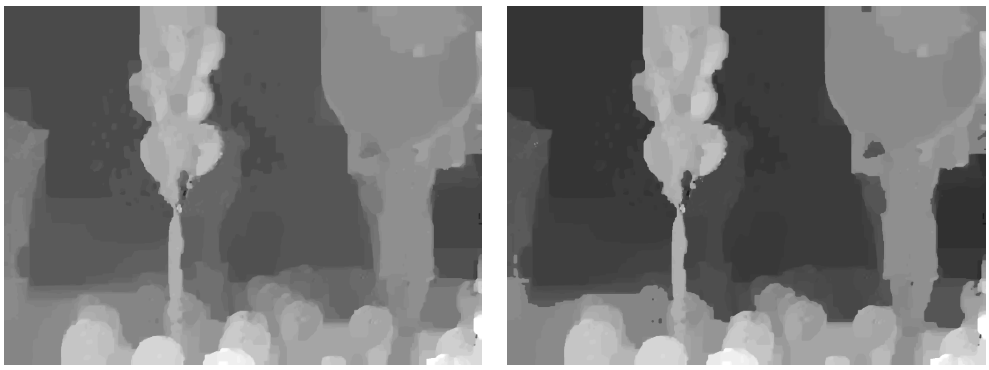
Pour l'image Balloons, les objets d'intérêt sont les ballons du premier plan et l'homme. Ainsi la zone de profondeur d'intérêt est limitée par $Z_{in\_l} = 128$ et $Z_{in\_h} = 255$. Pour l'image Undodancer, nous prenons l'exemple du premier plan limité par $Z_{in\_l} = 128$ et $Z_{in\_h} = 255$ et le plan contenant le danseur limité par $Z_{in\_l} = 100$ et $Z_{in\_h} = 121$. Le Th_{Quad} est choisi différemment afin d'avoir le même débit pour la grille sans et avec ajustement.

Les FIGURES 102 et 105 représentent la zone de profondeur d'intérêt DoI dans l'intervalle $[Z_{in\_l},\ Z_{in\_h}]$ dans la carte de profondeur originale d'entrée (masque de profondeur original d'entrée). Les FIGURES 103.a et 103.b représentent respectivement les cartes de profondeur originale et ajustée.

Les FIGURES 104 et 106 illustrent les résultats du QuadTree sans et avec ajustement de la dynamique de la carte de profondeur originale. Nous pouvons clairement remarquer que les contours des ballons de l'image *Balloons* et les contours des jambes du danseur de l'image *Undodancer* sont bien définis par rapport au QuadTree classique.

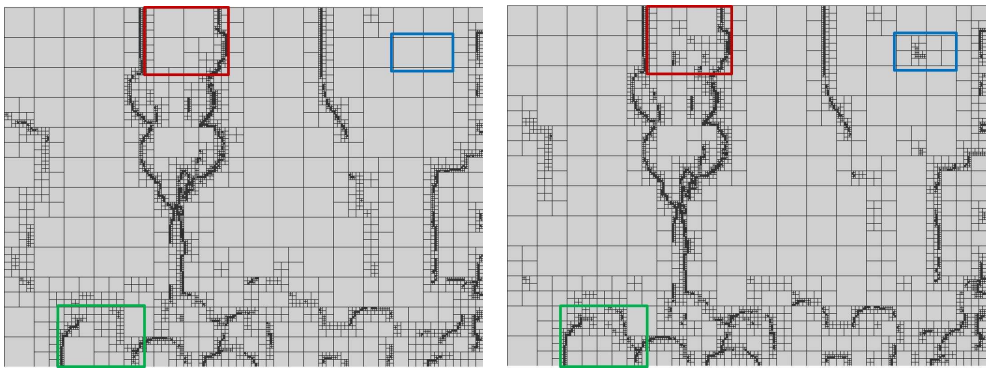
Après l'ajustement de la dynamique, l'activité à l'intérieur de la DoI est augmentée au détriment des zones de profondeur à l'extérieur de la DoI. En effet, à l'issue du QuadTree, la DoI est représentée par des blocs plus petits. La résolution locale y est ainsi incrémentée.

FIGURE 102. Masque binaire original d'entrée de Balloons vue 5 image 1 (1024x768 pixels) du premier plan avec $Z_{in\_l} = 128$, $Z_{in\_h} = 255$.



(a) (b)

FIGURE 103. Carte de profondeur de Balloons vue 5 image 1 : (a) originale ; (b) après ajustement de la dynamique du premier plan avec $Z_{in\_l} = 128$, $Z_{in\_h} = 255$, $F = 1.3$.

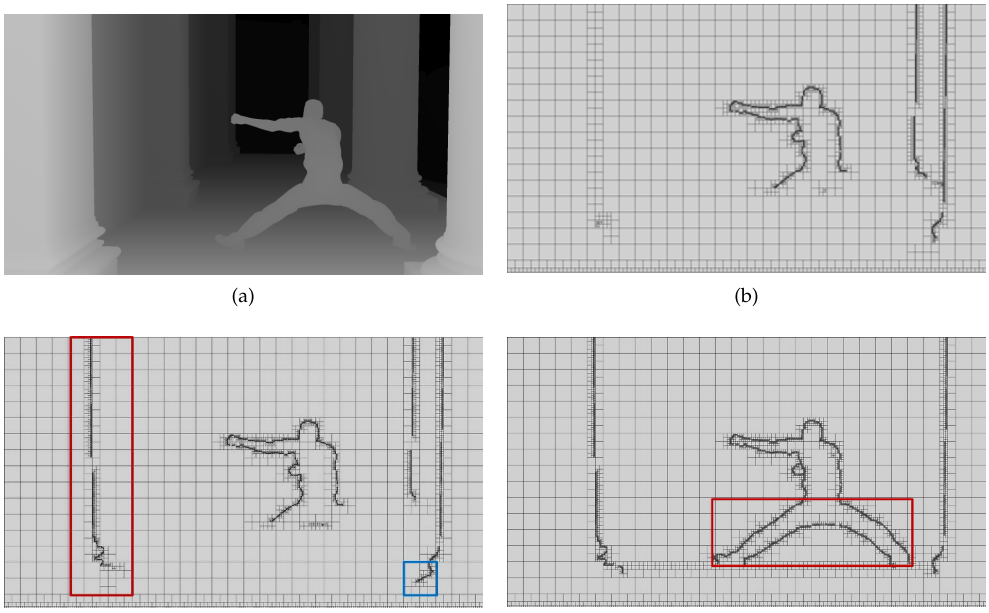


(a) (b)

FIGURE 104. Résultats de QuadTree de la carte de profondeur de Balloons vue 5 image 1 : (a) à partir de la carte originale avec $Th_{Quad} = 28$; (b) après ajustement de la dynamique du premier plan avec $Z_{in\_l} = 128$, $Z_{in\_h} = 255$, $F = 1.3$ et $Th_{Quad} = 29$.

(a) (b)

FIGURE 105. Masque binaire original d'entrée de Undodancer vue 1 image 250 (1920x1080 pixels) (a) du premier plan avec $Z_{in\_l} = 128$, $Z_{in\_h} = 255$; (b) de la zone de profondeur située entre $Z_{in\_l} = 110$ et $Z_{in\_h} = 121$.



(a) (b)

(c) (d)

FIGURE 106. Résultats de QuadTree de la carte de profondeur de Undodancer vue 1 image 250 : (a) carte de profondeur originale ; (b) QuadTree de la carte de profondeur originale avec $Th_{Quad} = 47$; (c) QuadTree de la carte de profondeur après ajustement de la dynamique du premier plan avec $Z_{in\_l} = 128$, $Z_{in\_h} = 255$, $F = 1.3$ et $Th_{Quad} = 60$; (d) QuadTree de la carte de profondeur après ajustement de la dynamique de la zone de profondeur située entre $Z_{in\_l} = 110$, $Z_{in\_h} = 121$, $F = 7.0$ et $Th_{Quad} = 71$.

6.2.5 *Codage de la DoI dans la texture*

Après le codage de la DoI dans la carte de profondeur, l'Autofocus 3D simple consiste ensuite en un codage de la DoI dans la texture. Ceci signifie que la texture doit être codée avec différentes qualités suivant la DoI donnée (voir FIGURE 100).

La première étape de codage DoI de la texture est l'extraction du masque binaire (Masque-DoI) qui va définir ultérieurement la Région d'Intérêt, considérant que seulement l'intervalle $[Z_{in\_l}, Z_{in\_h}]$ est transmis au décodeur. Cette extraction consiste simplement à binariser la carte de profondeur reconstruite, avec l'intervalle de profondeur $[Z_{in\_l}, Z_{in\_h}]$ comme paramètre d'entrée. La carte de profondeur considérée est celle qui est reconstruite, et ceci pour deux raisons principales. La première raison est que le processus doit être dupliqué au décodeur. La deuxième raison est que le masque doit être vu comme une partie de la partition QuadTree.

La seconde étape de codage DoI de la texture est le raffinement de qualité de la DoI dans la texture. Ce raffinement est exclusif aux objets se localisant dans la zone de profondeur d'intérêt. Nous introduisons deux approches pour le raffinement de qualité.

i. La première approche consiste à raffiner le SNR de la DoI en utilisant le concept de codage par Région d'Intérêt (RoI) introduit dans [77] : l'image est représentée par des régions adaptées au QuadTree, et chaque région est codée indépendamment avec un niveau de qualité SNR différent. La solution introduite dans [77], appelée LAR-RoI, permet de disposer de deux régions uniquement (RoI, Non-RoI). Nous étendons le LAR-RoI jusqu'à 8 niveaux de qualité, chaque niveau ayant son propre facteur de quantification, partant du label 0 pour la RoI. Le Masque-DoI extrait lors de la représentation de la DoI est utilisé ainsi pour définir la région d'intérêt. Le schéma proposé suppose au minimum trois facteurs de quantification : $Q_{p\_Z\_T\_BR}$ pour la profondeur et la texture à basse résolution, $Q_{p\_T\_Raf\_DoI}$ pour la DoI dans la texture raffinée, et $Q_{p\_T\_Raf\_NDoI}$ pour la texture raffinée à l'extérieur de la DoI (voir FIGURE 107). Dans ce chapitre, nous présentons uniquement les résultats pour cette configuration. Toutefois, d'autres scénarios sont possibles en exploitant le total des 7 niveaux de quantification possibles pour les zones non-DoI. Par exemple, un scénario simple consiste à diviser les zones non-DoI en N régions (N<8), puis d'appliquer une quantification adaptative de telle sorte que le facteur de quantification soit pondéré par la distance de la région à la DoI.

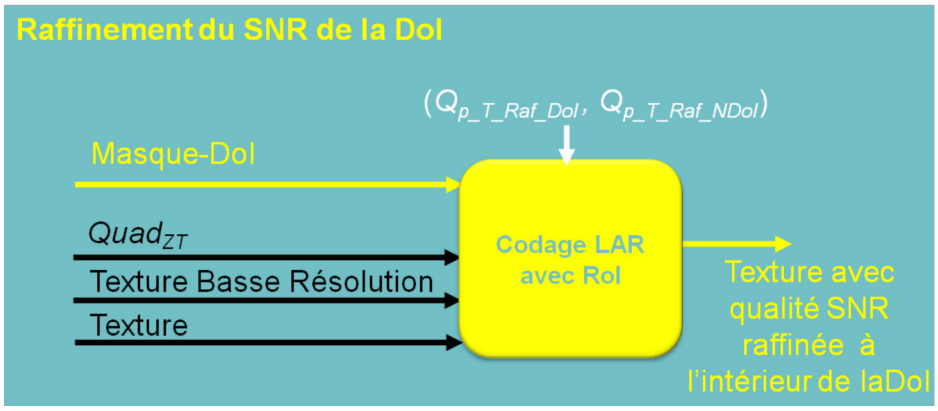


FIGURE 107. Codage par RoI basé sur le Masque-DoI.

ii. La seconde approche consiste à raffiner la résolution locale de la texture à l'intérieur de la DoI uniquement. Dans le schéma scalable proposé dans le Chapitre 4, le raffinement de la résolution s'applique sur toute la texture, en utilisant la grille profondeur plus texture $Quad_{ZT}$. Par contre, dans le cas présent où nous avons introduit la DoI, il s'agit de raffiner le $Quad_Z$ à l'intérieur de la DoI uniquement. Il consiste simplement à masquer l'estimation de la grille de profondeur plus texture $Quad_{ZT}$, par le Masque-DoI pour avoir une grille $Quad_{ZT\_DoI}$ (voir FIGURE 108).

Une solution jointe de raffinement du SNR et de résolution locale est également faisable.

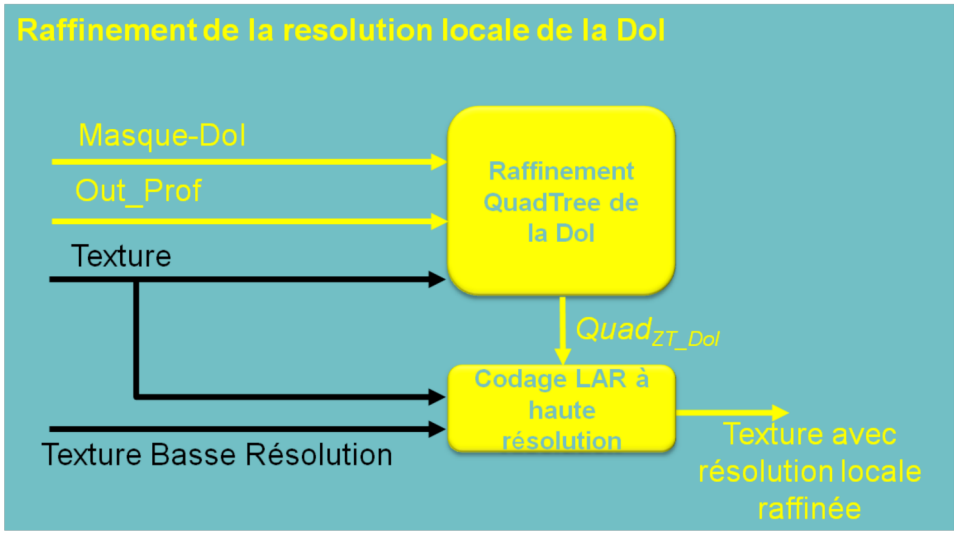


FIGURE 108. Raffinement de la résolution locale de la DoI uniquement.

6.2.6 *Expérimentation et Résultats d'Autofocus 3D*

Dans cette section, nous présentons les résultats d'Autofocus 3D simple. Les tests sont également effectués sur les séquences 3D de référence fournies par MPEG (images réelles : *Balloons*, *Kendo*, *BookArrival* et *Newspaper* de taille 1024x768, et images synthétisées par ordinateur : *UndoDancer* et *GTFly* de haute définition (1920x1080)). Nous présentons en premier lieu les résultats de codage DoI avec raffinement de qualité SNR de la texture, ensuite ceux avec raffinement de la résolution locale de la texture. Nous explorons enfin les résultats du schéma d'Autofocus 3D simple proposé sur les vues synthétisées. Les paramètres d'Autofocus 3D simple, Z_{in_l}, Z_{in_h} et le F-*number*, sont sélectionnés manuellement en fonction de la zone d'intérêt de l'image 3D.

Le schéma proposé de représentation et de codage est unique en termes de fonctionnalités combinées. Les comparaisons avec l'État de l'Art ne sont pas donc faisable. Toutefois, nous présentons dans ce qui suit des résultats comparatifs de codage RoI basé bloc au lieu de pixel. Plus de détails sur l'efficacité de compression 2D+Z comparée à l'État de l'Art, sont présentés dans le Chapitre 4.

6.2.6.1 *Résultats de codage DoI avec raffinement de qualité SNR de la texture*

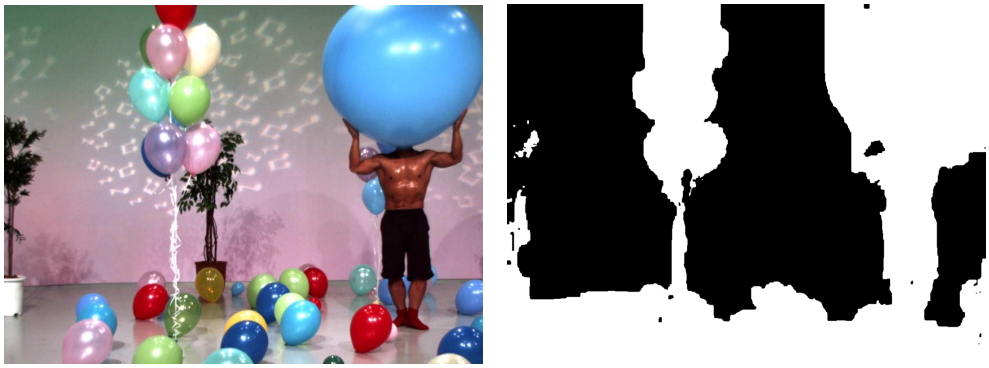
Quelques exemples de codage DoI avec raffinement de qualité SNR de la texture sont présentés dans cette sous-section. Les paramètres à régler sont :

- Z_{in_l}, Z_{in_h} pour la DoI,
- $Q_{p\_Raf\_T\_DoI}$ et $Q_{p\_Raf\_T\_NDoI}$ pour le raffinement.

Le critère de choix pour l'intervalle $[Z_{in\_l}, Z_{in\_h}]$ est le même que de la Section 6.2.4, et les facteurs de quantification sont choisis suivant la qualité désirée.

À titre d'exemple, les FIGURES 109 et 112 illustrent la texture originale, le masque de profondeur d'entrée original (défini à partir la carte de profondeur originale de la zone DoI entre Z_{in_l} et Z_{in_h}), et le Masque-DoI (défini à partir de la carte de profondeur reconstruite) avec différentes résolutions : masque à pleine résolution disponible pour le schéma proposé, et masque sous-échantillonné par blocs 8x8 ou 16x16 tel que l'on pourrait l'obtenir par un codeur classique de l'Ètat de l'Art, respectivement des images *Balloons* et *Undodancer*.

Les FIGURES 110 et 113 fournissent un zoom sur la qualité visuelle de la texture reconstruite après codage par le LAR classique et par l'Autofocus 3D simple, en utilisant le masque à pleine résolution et le masque sous-échantillonné. Les FIGURES 111 et 114 illustrent les régions d'intérêt extraites correspondantes. Nous pouvons clairement remarquer la netteté des contours des ballons et du danseur par le raffinement du SNR avec le masque à pleine résolution, en comparaison avec l'utilisation d'un masque à résolution en blocs 8x8 ou 16x16. La résolution au niveau pixel donne ainsi clairement une meilleure qualité visuelle.



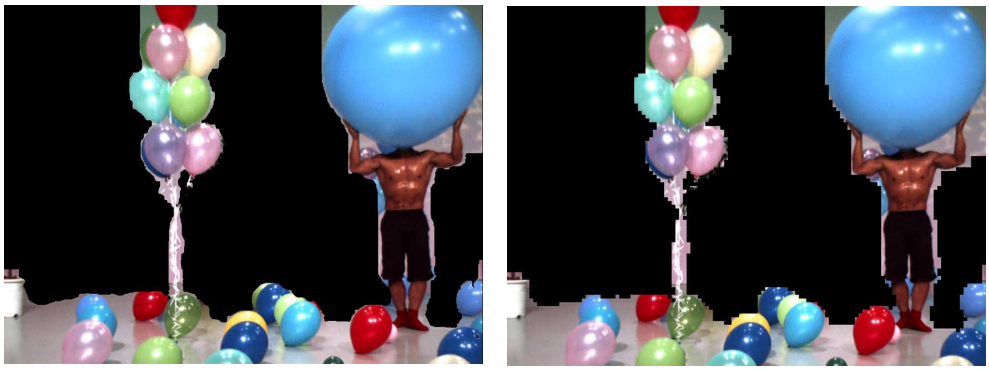
(a) Texture originale (b) Masque de profondeur original d'entrée

(c) Masque-DoI (d) Masque sous-echantillonné par bloc 8x8

FIGURE 109. Masque de profondeur d'intérêt pour le premier plan avec $Z_{in\_l} = 128$ et $Z_{in\_h} = 255$ de (a) Balloons vue 5 image 1 ; (b) extrait de la profondeur originale ; (c) extrait de la profondeur codée à 0.14 bpp ; (d) extrait de la profondeur codée, avec une résolution en blocs 8x8.

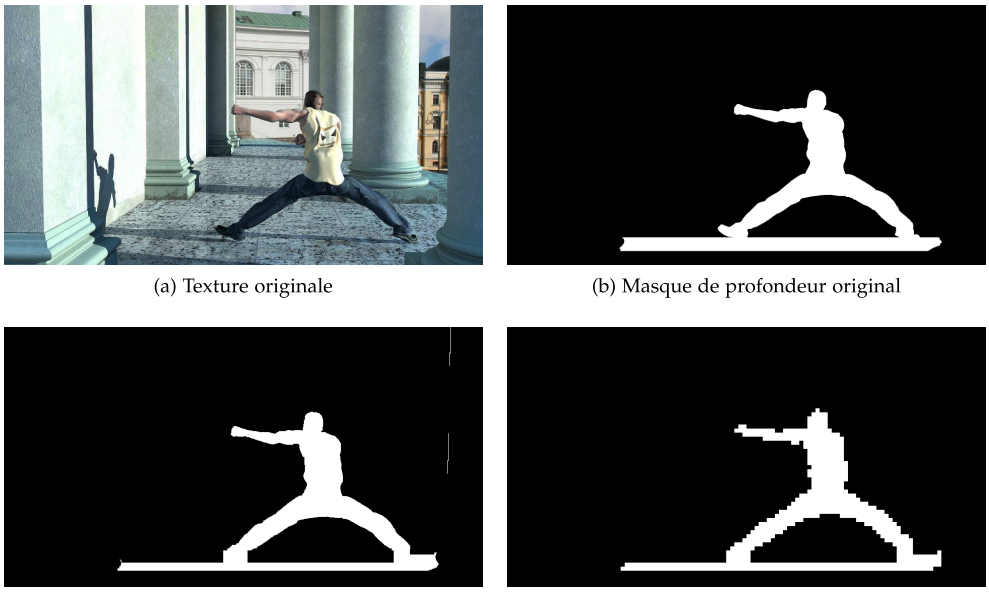
(a) (b) (c) (d)

FIGURE 110. Zoom sur la qualité visuelle de la texture de Balloons vue 5 image 1 (a) originale ; puis codée à 0.18 bpp (b) par le LAR classique (PSNR Global = 32.70 dB) ; (c) par raffinement de qualité SNR ($Q_{p\_Ref\_T\_DoI} = 25$, $Q_{p\_Ref\_T\_NDoI} = 120$) en utilisant le Masque-DoI à pleine résolution du premier plan avec $Z_{in\_l} = 128$ et $Z_{in\_h} = 255$ (PSNR Global = 33.05 dB, DoI codée à 0.54 bpp, $PSNR_{DoI} = 36.87$ dB ; Non-DoI codée à 0.12 bpp, $PSNR_{Non-DoI} = 30.74$ dB) ; (d) par raffinement de qualité SNR avec le Masque-DoI sous-échantillonné de résolution en blocs 8x8.



(a) (b)

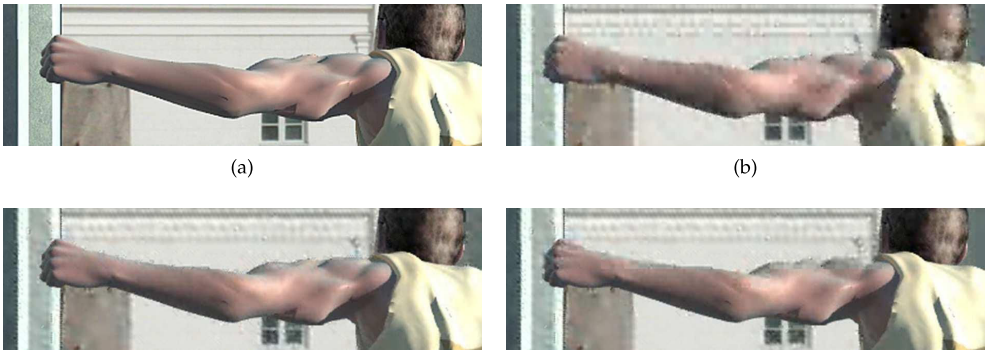
FIGURE 111. Région d'intérêt extraite de la texture codée par RoI en utilisant (a) le Masque-DoI à pleine résolution ; (b) Masque-DoI avec une résolution en blocs 8x8.



(a) Texture originale (b) Masque de profondeur original

(c) Masque-DoI (d) Masque-DoI sous-echantillonnee par bloc 16x16

FIGURE 112. Masque de profondeur d'intérêt pour le premier plan avec $Z_{in\_l} = 110$ et $Z_{in\_h} = 121$ de (a) Undodancer vue 1 image 250 ; (b) extrait de la profondeur originale ; (c) extrait de la profondeur codée à 0.07 bpp ; (d) extrait de la profondeur codée, avec une résolution en blocs 16x16.



(a) (b)

(c) (d)

FIGURE 113. Zoom sur la qualité visuelle de la texture de UndoDancer vue 1 image 250 (a) originale ; puis codée à 0.2 bpp (b) par le LAR classique (PSNR $Global = 24$ dB) ; (c) par raffinement de qualité SNR ($Q_{p\_Ref\_T\_DoI} = 25$, $Q_{p\_Ref\_T\_NDoI} = 120$) en utilisant le Masque-DoI à pleine résolution du premier plan avec $Z_{in\_l} = 110$ et $Z_{in\_h} = 121$ (PSNR $Global = 28.82$ dB, DoI codée à 0.98 bpp, $PSNR_{DoI} = 36.06$ dB ; Non-DoI codée à 0.17 bpp, $PSNR_{Non-DoI} = 28.1$ dB) ; (d) par raffinement de qualité SNR avec le Masque-DoI sous-échantillonné de résolution en blocs 8x8.

(a) (b)

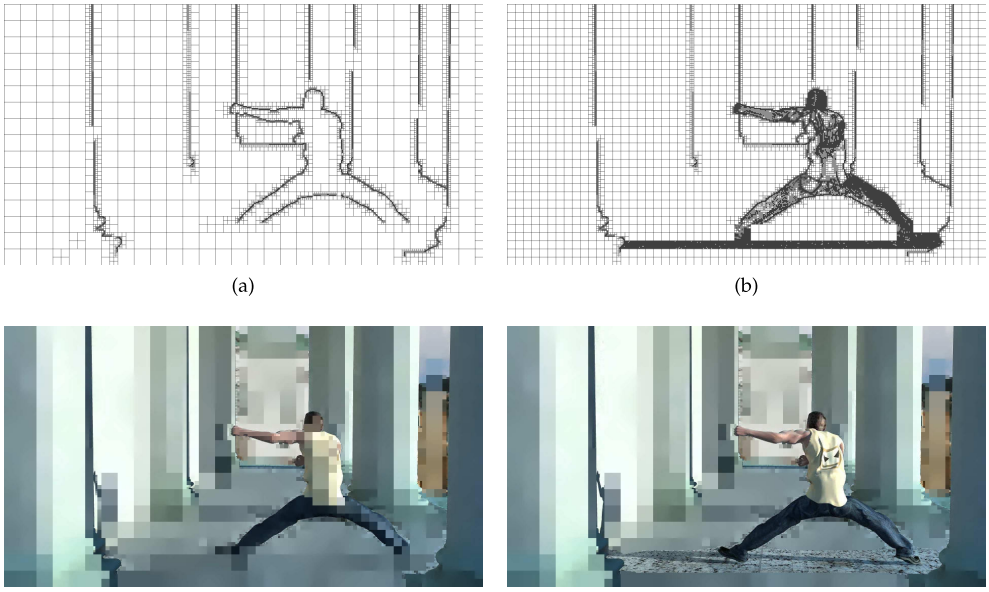
FIGURE 114. Région d'intérêt extraite de la texture codée par RoI en utilisant (a) le Masque-DoI à pleine résolution ; (b) le Masque-DoI avec une résolution en blocs 16x16.

6.2.6.2 *Résultats de codage DoI avec raffinement de la résolution locale de la texture*

Dans cette sous-section, nous présentons quelques exemples de codage DoI par raffinement de la résolution locale de la texture à l'intérieur de la DoI. En particulier, les FIGURES 115 et 116 illustrent la grille profondeur ($Quad_Z$), la grille de profondeur raffinée et masquée par le Masque-DoI ($Quad_{ZT-DoI}$), la texture basse résolution et la texture avec la résolution locale raffinée, respectivement de Newspaper et Undodancer. La FIGURE 115.e illustre un exemple de raffinement joint de qualité SNR et de résolution locale de la DoI. Une résolution plus fine est accordée à la texture au niveau de la DoI uniquement. Les qualités objectives et visuelles de la DoI sont ainsi améliorées.

(a) (b) (c) (d) (e) (f)

FIGURE 115. Comparaison de la qualité visuelle de la texture de Newspaper vue 6 image 1 avec $Th_{Quad} = 46$, $Q_p = 69$: (a) $Quad_Z$; (b) Masque-DoI ; (c) $Quad_{ZT\_DoI}$; (d) texture Basse Résolution à 0.04 bpp (PSNR Global = 18.2 dB) ; (e) texture avec la résolution locale raffinée à 0.1 bpp ($PSNR_{DoI} = 24.69$ dB) ; (f) texture avec raffinement de résolution locale et de qualité SNR ($Q_{p\_Ref\_T\_DoI} = 20$, $Q_{p\_Ref\_T\_NDoI} = 69$) à 0.31 bpp avec $Z_{in\_l} = 63$, $Z_{in\_h} = 255$, $F = 3$ ($PSNR_{DoI} = 26.52$ dB).



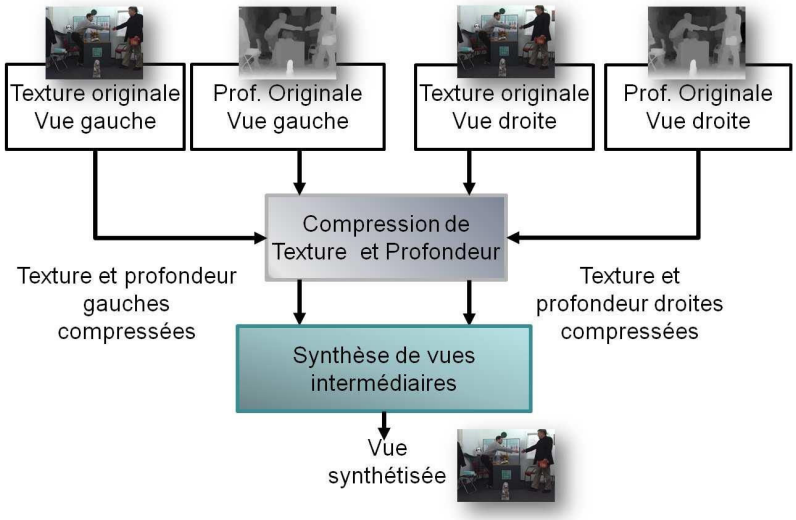
(a) (b)

(c) (d)

FIGURE 116. Comparaison de la qualité visuelle de la texture de UndoDancer vue 1 image 250 avec $Th_{Quad} = 20$, $Q_p = 30$: (a) $Quad_Z$; (b) $Quad_{ZT\_DoI}$; (c) texture Basse Résolution à 0.06 bpp (PSNR Global = 17.2 dB) ; (d) texture avec la résolution locale raffinée à 0.23 bpp avec $Z_{in\_l} = 110$, $Z_{in\_h} = 121$, $F = 7$ ($PSNR_{DoI} = 32.24$ dB).

6.2.6.3 *Résultats des vues synthétisées dans le contexte DoI*

Le dernier point important dans le codage 2D+Z est l'évaluation de la qualité visuelle des vues synthétisées. Pour la synthèse des vues intermédiaires, nous utilisons le logiciel *View Synthesis Reference Software (VSRS 3.0)* [26]. Dans cette série d'expériences, nous considérons les images de texture et de profondeur codées à bas débit sans et avec l'Autofocus 3D simple, afin d'évaluer l'effet de compression de la fonctionnalité proposée sur les vues synthétisées (voir FIGURE 117). Dans le but de comparer l'Autofocus 3D simple proposé avec les approches basées bloc (telles que H264 et HEVC), nous étudions l'effet de la résolution de la RoI sur les vues synthétisées en utilisant des textures codées avec les Masque-DoI à différentes résolutions : pleine résolution, résolution en blocs de 8x8 et 16x16.

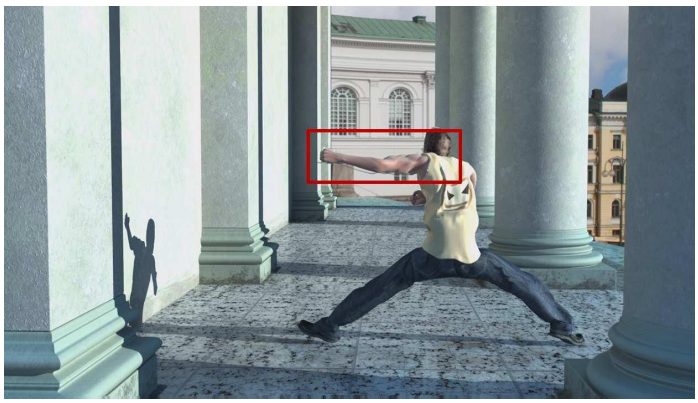


(a)

FIGURE 117. Plateforme de synthèse de vue intermédiaire.

La qualité objective (PSNR) ne peut pas être calculée pour les vues synthétisées puisqu'il n'existe ni une image synthétisée de référence ni un masque de référence. Pourtant, nous pouvons remarquer sur la FIGURE 118, que la qualité de la DoI dans les vues synthétisées à partir des cartes de profondeur codées avec l'Autofocus 3D simple est mieux que celles synthétisées à partir des cartes codées par le LAR classique avec le même débit (voir FIGURE 118.c, 118.d). En outre, les images de texture codées avec le schéma lui même basé bloc produisent des vues synthétisées à faible qualité, et plus particulièrement sur les contours de la DoI. Les images de texture codées avec le schéma de codage DoI basé pixel proposé conduit à une qualité plus fine et des contours plus précis au niveau de la DoI dans les vues synthétisées (voir FIGURE 118).

L'Autofocus 3D simple proposé assure ainsi une forte consistance entre la texture et la profondeur et une haute qualité sur les contours des objets dans la texture et la profondeur ainsi que dans les vues synthétisées.



(a)

(b)

(c)

(d)

(e)

FIGURE 118. Comparaison de la qualité visuelle de la vue synthétisée de (a) Undodancer vue 3 image 250 b) en utilisant les profondeur et texture originales ; en utilisant la profondeur reconstruite à 0.014 bpp et texture reconstruite à 0.2 bpp c) par le LAR classique ; d) avec l'Autofocus 3D par raffinement de qualité SNR de la texture avec le Masque-DoI à pleine résolution ; e) avec l'Autofocus 3D par raffinement de qualité SNR de la texture avec le Masque-DoI à une résolution en bloc 16x16.

6.3 SCHÉMA GLOBAL D'AUTOFOCUS 3D AVANCÉ BASÉ SEGMENTATION SÉMANTIQUE

La deuxième solution proposée pour coupler un schéma de représentation pixelique avec un schéma de codage basé bloc est présentée dans cette partie du chapitre. Il s'agit d'un schéma d'"Autofocus 3D avancé" basé sur une segmentation sémantique 3D. Comme illustré dans la FIGURE 119, le schéma joint de segmentation sémantique et d'Autofocus 3D est intégré dans le schéma de codage global 2D+Z proposé dans le Chapitre 4.

Dans un premier temps, nous proposons un algorithme de segmentation sémantique basé sur la version à basse résolution de l'image 3D. Le terme sémantique se réfère à la possibilité d'ajuster la segmentation en régions de l'image suivant une "pondération" ou " ajustement de curseur" entre l'image 2D et la profondeur défini par l'utilisateur (voir FIGURE 120) : si la

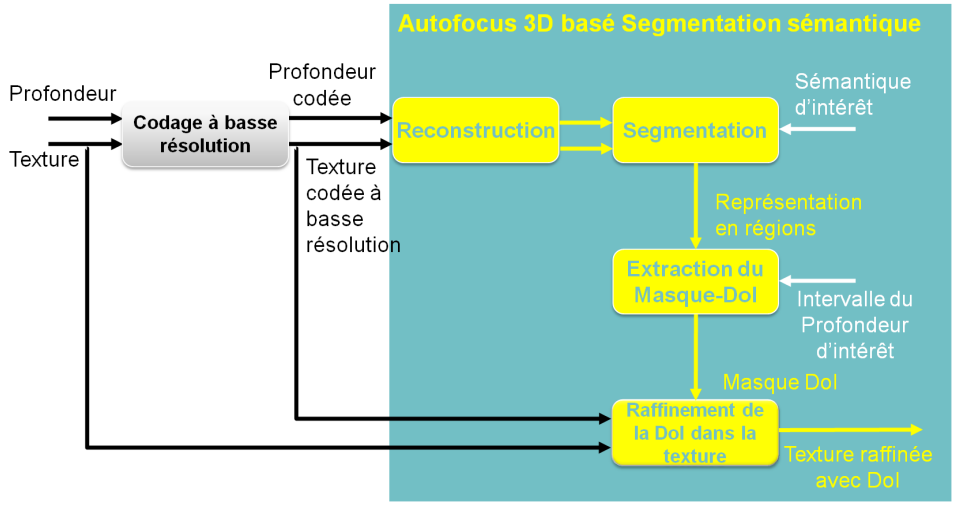


FIGURE 119. Schéma joint de segmentation sémantique et d'Autofocus 3D.

couleur est seule prise en compte pour la segmentation, les objets ayant des couleurs différentes doivent donc être segmentés dans des régions distinctes (voir FIGURE 120.a). Au contraire, si la profondeur est celle prise en compte toute seule, alors les objets appartenant au même plan de profondeur doivent appartenir à une seule région, même si leurs couleurs sont différentes (voir FIGURE 120.b). Une pondération entre la couleur et la profondeur peut être utilisée afin d'obtenir une segmentation convenable (voir FIGURE 120.c). La segmentation sémantique 3D est une extension de l'algorithme de segmentation 2D de *Deforges et al.* [77].

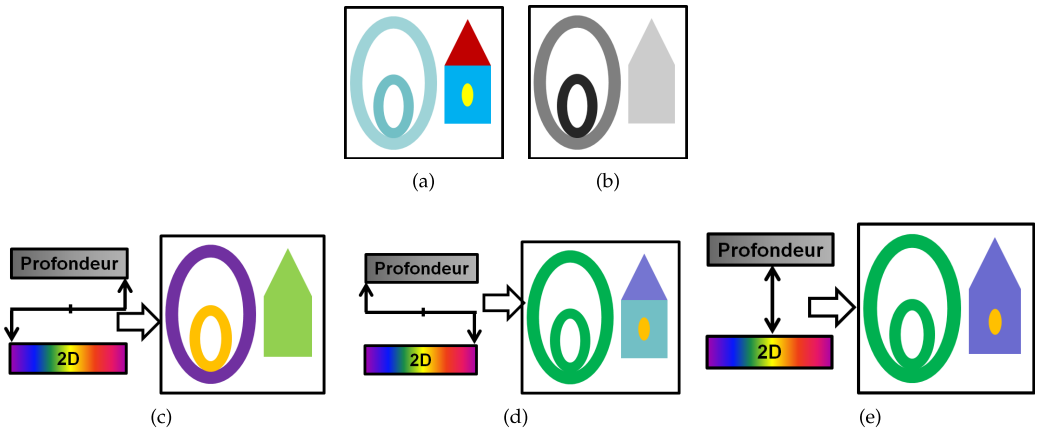


FIGURE 120. Segmentation sémantique suivant une pondération (ajustement de curseur) entre (a) la texture et (b) la profondeur : (c) segmentation suivant la texture uniquement ; (d) segmentation suivant la profondeur uniquement ; (e) segmentation suivant une balance entre la profondeur et la texture.

A l'issue de l'étape de segmentation, il est possible d'appliquer simplement le même principe d'Autofocus 3D simple précédent, pour une extraction et un codage de la profondeur d'Intérêt (DoI).

Ainsi dans ce qui suit de ce chapitre, la Section 6.3.1 décrit l'algorithme de segmentation 2D de *Deforges et al.*. La Section 6.3.2 introduit ensuite l'algorithme de segmentation sémantique 3D. Enfin, la Section 6.3.3 présente la procédure d'Autofocus 3D avancé basé sur la représentation en régions.

6.3.1 *Segmentation 2D de Deforges et al.*

Deforges et al. proposent dans [77], une technique de représentation en régions auto-extractibles à coût nul. Pour éviter le surcoût de codage de cartes des régions, l'algorithme de segmentation est réalisé au codeur et décodeur à partir de l'image à bas débit. La partition initiale de l'image correspond ainsi à la représentation basée QuadTree, ce qui assure notamment une représentation cohérente entre contours et zones uniformes, même à très bas débits. L'algorithme de segmentation 2D de *Deforges et al.* peut être comparé à la technique Split & Merge qui consiste à découper l'image en blocs, puis les fusionner pour former les régions. Dans le cas du LAR, l'image est donc déjà découpée par le QuadTree. L'algorithme de segmentation est par suite réduit à la phase de fusion. Il faut noter que le processus de segmentation opérant directement au niveau des blocs, il est très rapide comparé à des techniques analogues appliquées au niveau pixel.

6.3.1.1 *Description de la segmentation 2D*

Soit une image I de taille $(N_x \times N_y)$. On introduit les notations suivantes :

- QT : partitionnement QuadTree de l'image I en P blocs carrés B_i, $i \in \{1...P\}$, avec chaque bloc B_i de surface $2^N \times 2^N$, $N \in 1...N_{max}$.
- la partition QT couvre l'image I sans recouvrement :
  $QT = \bigcup_{j=1}^{P} B_j$, avec $B_j \bigcap B_i = \emptyset \quad \forall (i,j) \in \{1...P\}^2 \quad i \neq j$.
- Δ^K : partition de l'image avec K régions.
- Δ^{K_0} : partition initiale de l'image I avec K_0 régions ($\Delta^{K_0} \equiv QT$).
- G^K : graphe d'adjacence non orienté correspondant à la partition Δ^K.
- R_k^K : région numéro k dans la partition Δ^K.
- $Surf(R_k^K)$: surface en pixels de la région R_k^K.
- A_i^K : ensemble des régions connexes à R_i^K dans la partition Δ^K.

Le but du processus de segmentation est de passer de la partition initiale Δ^{K_0} qui est le QuadTree QT à une nouvelle partition Δ^K ($K < K_0$) selon un critère d'homogénéité et à travers des séquences de fusions de régions $\Delta^K = \bigcup_{k=1}^{K} R_k^K$.

Le processus de segmentation est donc initialisé par le QuadTree ($\Delta^{K_0} = QT$). Le passage vers une nouvelle partition Δ^K consiste ensuite à fusionner les régions connexes selon un critère d'homogénéité. Le critère de fusion largement utilisé dans les méthodes de segmentation est le coût minimal entre deux régions. Un tel type de solutions est généralement considéré comme optimal, mais il nécessite un calcul important, comme mentionné dans le Chapitre 5. Afin de réduire le temps du processus de segmentation, *Deforges et al.* utilisent le critère de fusion basé sur un seuil : le coût de segmentation est tout d'abord calculé entre une région et ses régions adjacentes, une opération de fusion est ensuite permise si le minimum de ces coûts est inférieur à un seuil déterminé (Th_{Cost}). Le coût utilisé est : 1)pondéré et 2) basé moyenne et gradient.

1) Coût pondéré. Le coût défini entre deux régions est pondéré par la surface de la région, de sorte que si $Cost(R_i^K, R_j^K)$ définit le coût entre deux régions, alors le coût pondéré $Cost'(R_i^K, R_j^K)$ est donné par :

$$Cost'(R_i^K, R_j^K) = Cost(R_i^K, R_j^K) log_{10}\left(Surf\left(R_i^K\right)\right) \tag{35}$$

2) Coût basé moyenne et gradient. -Le coût moyen $Cost_M(R_i^K, R_j^K)$ est construit à partir de la différence des valeurs moyennes de régions (voir Eq. 36). Il permet de juger le degré d'homogénéité des régions d'un point de vue couleur.

$$Cost_M(R_i^K, R_j^K) = \left| \left[R_i^K\right] - \left[R_j^K\right] \right|, \tag{36}$$

avec $\left[R_i^K\right]$ désigne la valeur moyenne de la région R_i^K.

-Dans des zones uniformes comportant un gradient local, la segmentation basée uniquement sur le coût moyen ne fonctionne pas correctement et génère des "faux contours". À ce coût moyen, *Deforges et al.* associent donc un coût $Cost_{Gr}$ basé sur le calcul du gradient sur les frontières entre les régions.

Finalement, la coût total ($Cost(R_i^K, R_j^K)$) entre deux régions est considéré comme la moyenne des deux coûts : moyen et gradient (voir Eq. 37).

$$Cost(R_i^K, R_j^K) = \frac{Cost_M(R_i^K, R_j^K) + Cost_{Gr}(R_i^K, R_j^K)}{2} \tag{37}$$

Pour chaque balayage du graphe d'adjacence, l'algorithme de *Deforges et al.* détermine simplement pour chaque région, la région la plus proche d'un point de vue coût, puis la fusion est effectuée si le coût est inférieur au seuil Th_{Cost}. Le processus est réitéré jusqu'à ce qu'aucune fusion ne soit obtenue.

Le processus de segmentation 2D de *Deforges et al.* est décrit par l'algorithme 6.3.1.

Algorithme 6.3.1 *: Segmentation 2D*

Δ^{K_0} : partition initiale (blocs)
$Nb_{fusions} = 0; K = K_0;$
Faire
  $Nb_{fusions\_prec} = Nb_{fusions}; i = 1;$
  Faire
    Si $R_i^K \in G^K$

      Trouver $R_j^K \in A_i^K$ tel que $Cost(R_i^K, R_j^K) \leqslant Cost(R_i^K, R_l^K), \forall\, R_l^K \in A_i^K$

    Fin si ;
    Incrémenter i;
  Tant que $i \leqslant K_0$;

  $i = 1;$
  Faire
    Si $R_i^K \in G^K$

      Si $Cost'(R_i^K, R_j^K) < Th_{Cost}$

        Fusionner R_i^K et R_j^K;

        $K = K - 1$; Incrémenter $Nb_{fusions}$;
      Fin Si ;
    Fin Si ;
    Incrémenter i;
Tant que $i \leqslant K_0$;
Tant que $Nb_{fusions\_prec} < Nb_{fusions}$

L'algorithme de segmentation 2D de *Deforges et al.* possède deux points forts par rapport à d'autres algorithmes :

- il opère au niveau des blocs au lieu des pixels ce qui le rend plus rapide,
- il pondère le critère de segmentation avec la surface des régions, ce qui évite la sur-segmentation.

Grâce à sa faible complexité et ses très bonnes performances, nous avons proposé une extension 3D à l'algorithme de segmentation 2D de *Deforges et al.*, que nous présentons dans la sous-section suivante.

6.3.2 *Segmentation sémantique 3D*

Dans le domaine 2D, la segmentation est basée uniquement sur la couleur de la scène (les composantes RGB ou YCbCr, ...). La représentation en régions peut ainsi manquer de cohérence avec la réelle disposition des objets dans la scène. Ceci peut se traduire notamment par des phénomènes de sur-segmentation ou sous-segmentation de la scène.

Le représentation 3D, classiquement sous la forme 2D+Z permet alors une représentation plus discriminante et facilement exploitable, à travers l'information de distance des objets. Nous détaillons ainsi dans cette section l'extension au domaine 3D de l'algorithme de segmentation 2D initial de *Deforges et al.*. Une telle extension exploite la carte de profondeur comme une information sémantique additionnelle afin d'améliorer la segmentation en régions.

6.3.2.1 *Description de l'algorithme de segmentation sémantique 3D*

Le processus de segmentation 2D+Z est basé sur l'algorithme de segmentation 2D proposé par *Deforges et al.* et expliqué dans la section 6.3.1. Le processus de segmentation nécessite : 1) l'image de texture, 2) la profondeur et 3) le QuadTree. Néanmoins, plusieurs choix existent :

- le QuadTree basé profondeur seule, texture seule, ou les deux en même temps ?
- la texture est prise à basse ou à haute résolution ?

Dans le contexte de compression à bas débit et d'une représentation en régions non fine, le QuadTree basé profondeur seule ($Grille_{Prof}$) est préféré, avec la texture basse résolution (Tex_{BR}) et la profondeur pour la même partition. Par contre, dans le contexte de compression à haut débit, le QuadTree basé profondeur plus texture ($Grille_{Prof+Tex}$) sera utilisé avec la texture à haute résolution (Tex_{HR}) et la profondeur.

La sémantique d'intérêt, en terme de 2D/profondeur, dépend de l'application. Nous introduisons cette sémantique dans le processus de segmentation sous forme de paramètres d'entrée pour la luminance (Y), les chrominances (C_b, C_r) et la profondeur (Z). Le coût de segmentation est ensuite calculé en utilisant un ajustement pondéré entre la texture et la profondeur suivant l'équation (38). Cette pondération contrôle ainsi le coût de segmentation, afin d'avoir une représentation interprétable de la scène (voir FIGURE 120).

$$\begin{aligned} Cost_M &= \alpha \times Cost_M(Y) + \beta_1 \times Cost_M(Cb) + \beta_2 \times Cost_M(Cr) + \gamma \times Cost_M(Z) \\ Cost_{Gr} &= \alpha \times Cost_{Gr}(Y) + \beta_1 \times Cost_{Gr}(Cb) + \beta_2 \times Cost_{Gr}(Cr) + \gamma \times Cost_{Gr}(Z) \end{aligned} \tag{38}$$

avec α, β_1, β_2 et γ les coefficients d'entrée de pondération représentant la sémantique d'intérêt.

Le coût de segmentation est calculé entre une région et ses régions adjacentes. Ensuite une opération de fusion est permise si le minimum de ces coûts est inférieur au seuil déterminé (Th_{Cost}). Ce paramètre contrôle ainsi la granularité de la représentation finale en régions : plus Th_{Cost} est élevé, moins nous obtenons un nombre de régions dans la représentation finale, puisqu'il y aura beaucoup plus de régions fusionnées.

L'image de profondeur contient généralement moins d'activité que la luminance de la texture. La conséquence en est que pour un même seuil Th_{Cost}, le nombre de régions issues de la profondeur toute seule va être très petit en comparaison avec la segmentation issue de la texture uniquement. Pour résoudre ce problème et pour avoir un niveau comparable de représentation, nous avons introduit une solution d'un seuil adaptatif. Considérant que l'augmentation de débit demande un seuil de fusion plus élevé, le débit de compression relatif (R_k) de chaque composante est intégré. Ainsi, pour un $Th_{SegInput}$, nous calculons l'expression du nouveau seuil Th_{Cost} suivant l'équation (39) :

$$\begin{aligned} Th_{Cost} &= \frac{\alpha.R_Y + \beta_1.R_{Cb} + \beta_2.R_{Cr} + \gamma.R_Z}{\alpha + \beta_1 + \beta_2 + \gamma} Th_{SegInput} \\ &\text{où } R_k = \frac{débit_k}{débit_Z};\ k = Y,\ C_b,\ C_r,\ Z; \end{aligned} \tag{39}$$

Nous pouvons ainsi avoir une représentation interprétable de la scène par contrôle de la granularité de la segmentation :

- contrôler le coût de segmentation (ajustement 2D/Profondeur),
- contrôler le seuil de segmentation (seuil en fonction de l'ajustement 2D/Profondeur).

6.3.2.2 *Expérimentation et Résultats de la segmentation sémantique 3D*

Les tests sont réalisés sur plusieurs images 3D de référence fournies par MPEG, *Balloons*, *Kendo*, *BookArrival*, *Newspaper* de taille 1024x768 et *UndoDancer* de taille 1920x1080. Dans ces images, un seul objet peut contenir plusieurs couleurs, d'autres objets du premier plan ont la même couleur du fond. Ceci permet d'évaluer les performances de l'algorithme proposé dans des situations où l'objet risque d'être sur-segmenté ou fusionné avec le fond.

La segmentation 3D proposée a été appliquée en considérant les différents paramètres :

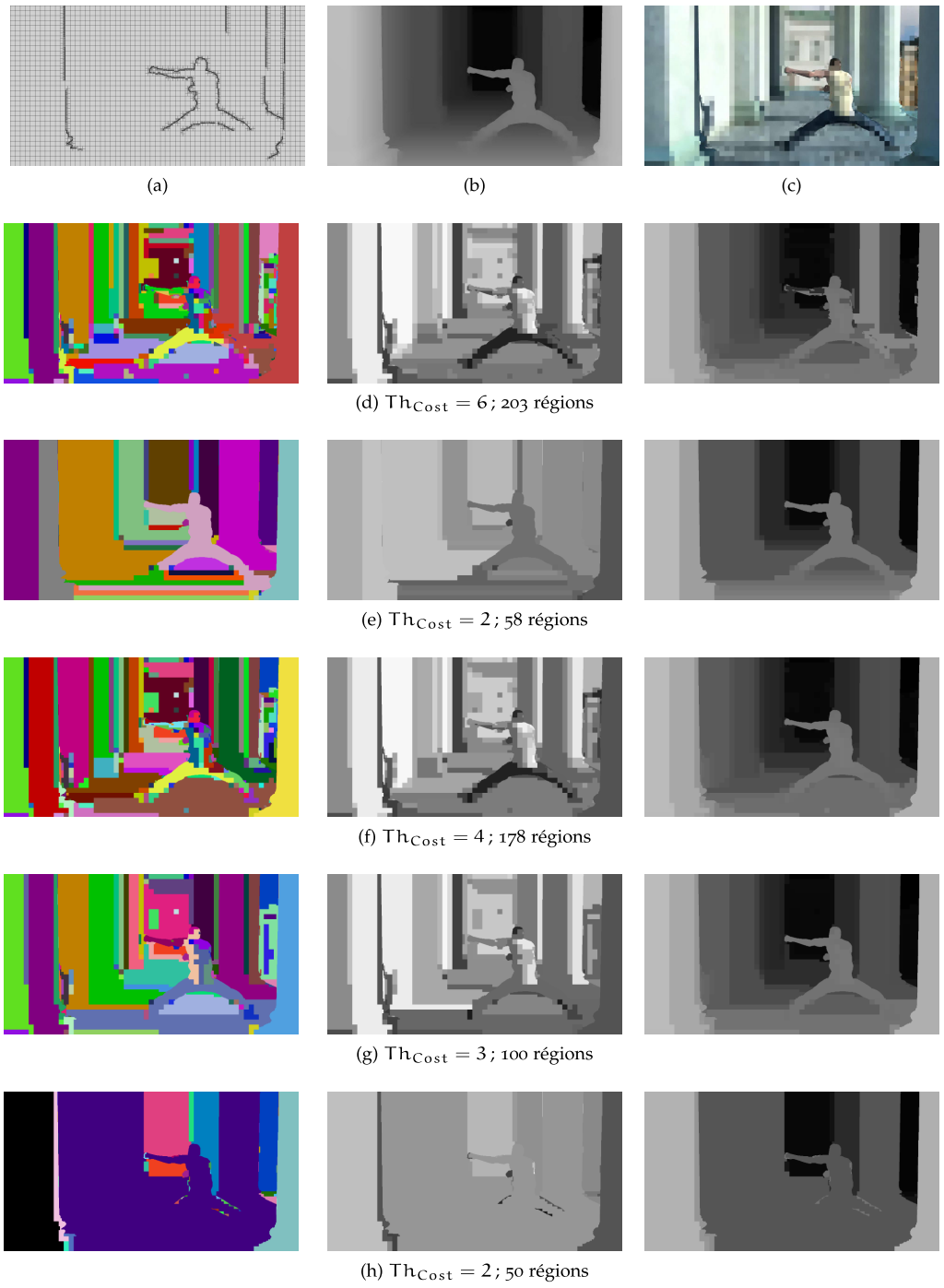
- Th_{Quad} pour la qualité de l'image 2D+Z compressée,
- $(\alpha, \beta_1, \beta_2, \gamma)$ pour le choix de la sémantique entre 2D et Z,
- $Th_{SegInput}$ pour le critère de fusion,
- et finalement le choix entre le QuadTree basé profondeur uniquement ou profondeur et texture.

Pour les résultats montrés dans cette section, nous avons choisi à titre d'exemple $Th_{Quad} = 33$, avec différents $Th_{SegInput}$ et différentes combinaisons des coefficients de pondération.

Les FIGURES 121, 122, 123 et 124 illustrent les résultats pour l'exemple de *Undodancer* dans les contextes de bas débit et de haut débit. Dans les figures 121 et 122, le processus de segmentation débute à partir du QuadTree basé profondeur ($Quad_Z$) uniquement, ce qui accélère la segmentation. Les résultats de segmentation sont présentés par colonne en considérant une représentation en régions par fausses couleurs, par moyenne de la luminance et par moyenne de la profondeur. Dans la FIGURE 121.d, l'image d'*Undodancer* est partitionnée en 285 régions, tandis que dans la FIGURE 121.e, où la profondeur est uniquement considérée, l'image est partitionnée en 60 régions seulement. Les FIGURES 121.f et 121.g montrent un bon compromis entre la profondeur et la texture.

Les FIGURES 123 et 124 illustrent un exemple dans un contexte à plus haut débit en considérant le $Quad_{ZT}$ avec une représentation plus fine. Dans la Figure 123.d, où la texture à haute résolution (Tex_{HR}) est uniquement considérée, l'image est partitionnée en un grand nombre de régions et le danseur est partiellement fusionné avec le fond. Deux exemples, avec une texture à haute résolution et une profondeur à haute résolution, sont ensuite illustrés dans les FIGURES 123.e et 123.f. Nous pouvons remarquer que l'image est partitionnée en un nombre moindre de régions, et possède une représentation plus fidèle et plus interprétable.

Cette approche présente plusieurs intérêts : 1) il est possible d'obtenir la représentation en régions appropriée à partir des images visuellement très compressées, par un ajustement entre l'image 2D et la profondeur, en fonction de la sémantique d'intérêt de l'utilisateur ou de l'application ; 2) il est possible de choisir entre un processus de segmentation avec une représentation détaillée de la scène et un processus rapide avec une représentation globale de la scène ; 3) quelles que soient les configurations sémantiques, la méthode de représentation en régions proposée conserve les principaux contours des objets avec une précision au niveau pixel, et maintient la cohérence spatiale entre la profondeur et la texture de la scène.



(a) (b) (c)

(d) $\mathrm{Th}_{\mathrm{Cost}} = 6$; 203 régions

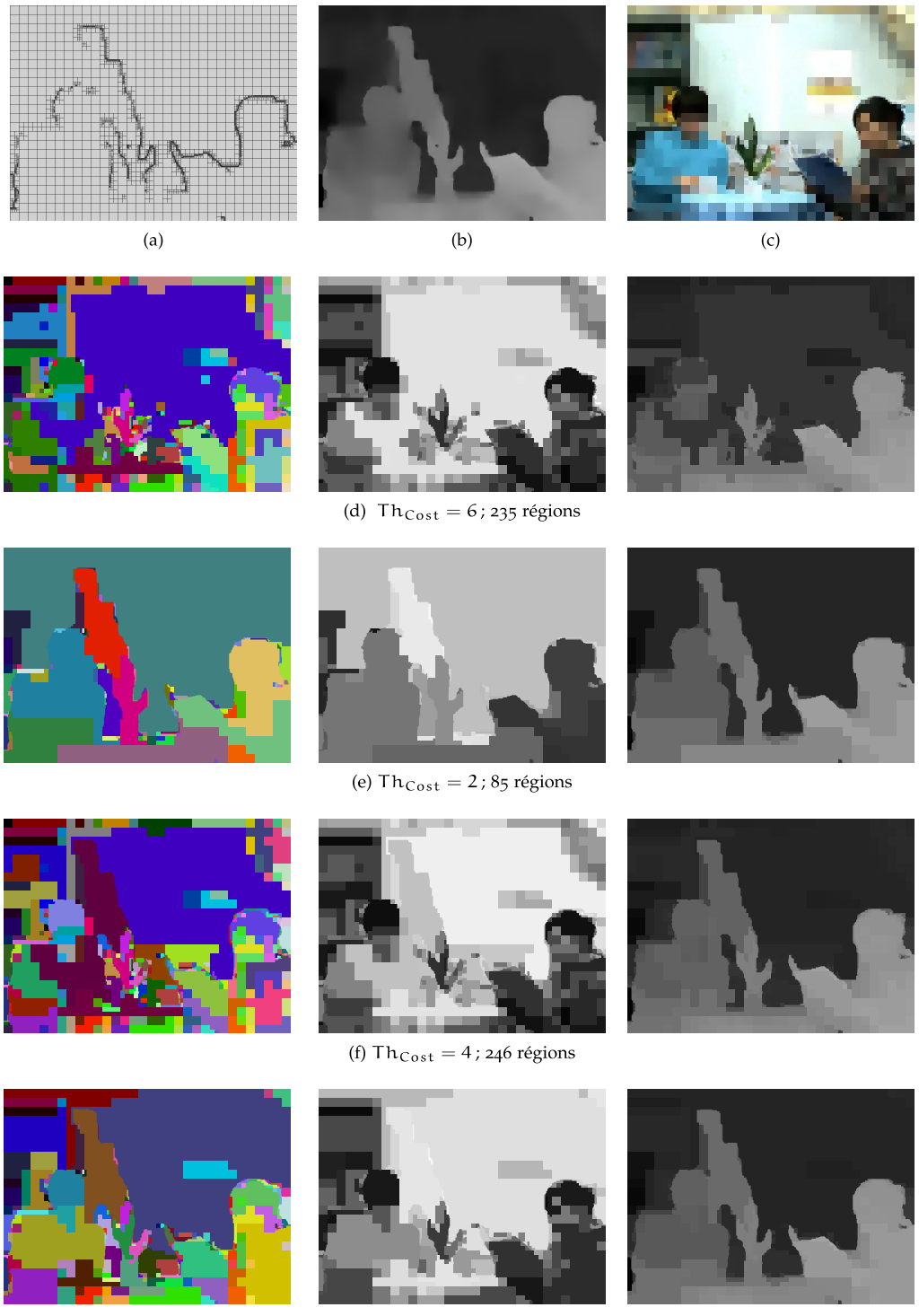
(e) $\mathrm{Th}_{\mathrm{Cost}} = 2$; 58 régions

(f) $\mathrm{Th}_{\mathrm{Cost}} = 4$; 178 régions

(g) $\mathrm{Th}_{\mathrm{Cost}} = 3$; 100 régions

(h) $\mathrm{Th}_{\mathrm{Cost}} = 2$; 50 régions

FIGURE 121. Résultats de segmentation à bas débit de Undodancer vue 1 image 250 utilisant (a) Quad_Z à 0.003 bpp ; (b) profondeur reconstruite (0.017 bpp, PSNR = 38 dB) et (c) texture reconstruite (0.04 bpp, PSNR = 17 dB). Segmentation basée d) texture ($\alpha = 1, \beta_1 = \beta_2 = \gamma = 0$) ; (e) profondeur ($\alpha = \beta_1 = \beta_2 = 0, \gamma = 1$) ; (f) 50% texture et 50% profondeur ($\alpha = 0.5, \beta_1 = \beta_2 = 0, \gamma = 0.5$) ; (g) 80% profondeur et 20% texture ($\alpha = 0.2, \beta_1 = \beta_2 = 0, \gamma = 0.8$) ; (h) 50% profondeur, 25% Cb, et 25% Cr ($\alpha = 0, \beta_1 = \beta_2 = 0.25, \gamma = 0.5$).



(a) (b) (c)

(d) $\mathrm{Th}_{\mathrm{Cost}} = 6$; 235 régions

(e) $\mathrm{Th}_{\mathrm{Cost}} = 2$; 85 régions

(f) $\mathrm{Th}_{\mathrm{Cost}} = 4$; 246 régions

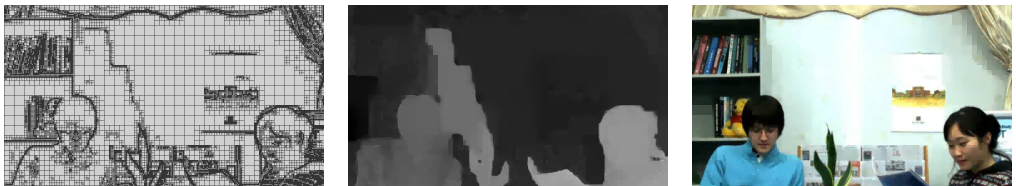
(g) $\mathrm{Th}_{\mathrm{Cost}} = 3$; 159 régions

FIGURE 122. Résultats de segmentation à bas débit de Newspaper vue 6 image 1 utilisant (a) Quad_Z à 0.004 bpp ; (b) profondeur reconstruite (0.031 bpp, PSNR = 36.75 dB) et (c) texture reconstruite (0.04 bpp, PSNR = 18.26 dB). Segmentation basée d) texture ($\alpha = 1, \beta_1 = \beta_2 = \gamma = 0$) ; (e) profondeur ($\alpha = \beta_1 = \beta_2 = 0, \gamma = 1$) ; (f) 50% texture et 50% profondeur ($\alpha = 0.5, \beta_1 = \beta_2 = 0, \gamma = 0.5$) ; (g) 80% profondeur et 20% texture ($\alpha = 0.2, \beta_1 = \beta_2 = 0, \gamma = 0.8$).

(a) (b) (c)

(d) $\mathrm{Th}_{\mathrm{Cost}} = 8$; 867 régions (e) $\mathrm{Th}_{\mathrm{Cost}} = 5$; 546 régions (f) $\mathrm{Th}_{\mathrm{Cost}} = 2$; 589 régions

FIGURE 123. Résultats de segmentation à haute résolution de Undodancer vue 1 image 250 utilisant (a) Quad_{ZT} à 0.04 bpp ; (b) profondeur reconstruite (0.03 bpp, PSNR = 44.26 dB) et c) texture reconstruite (0.41 bpp, PSNR = 31.95 dB). Segmentation basée d) texture ($\alpha = 1, \beta_1 = \beta_2 = \gamma = 0$) ; (e) 50% texture et 50% profondeur ($\alpha = 0.5, \beta_1 = \beta_2 = 0, \gamma = 0.5$) ; (f) 80% profondeur et 20% texture ($\alpha = 0.2, \beta_1 = \beta_2 = 0, \gamma = 0.8$).



(a) (b) (c)

(d) $\mathrm{Th}_{\mathrm{Cost}} = 8$; 292 régions (e) $\mathrm{Th}_{\mathrm{Cost}} = 5$;213 régions (f) $\mathrm{Th}_{\mathrm{Cost}} = 2$; 317 régions

FIGURE 124. Résultats de segmentation à haute résolution de Newspaper vue 6 image 1 utilisant (a) Quad_{ZT} à 0.031 bpp ; (b) profondeur reconstruite (0.05 bpp, PSNR = 38.49 dB) et (c) texture reconstruite (0.18 bpp, PSNR = 30.06 dB). Segmentation basée d) texture ($\alpha = 1, \beta_1 = \beta_2 = \gamma = 0$) ; (e) 50% texture et 50% profondeur ($\alpha = 0.5, \beta_1 = \beta_2 = 0, \gamma = 0.5$) ; (f) 80% profondeur et 20% texture ($\alpha = 0.2, \beta_1 = \beta_2 = 0, \gamma = 0.8$).

6.3.3 *Autofocus 3D avancé basé segmentation sémantique*

Nous adoptons le même principe d'Autofocus présenté dans la première partie du chapitre. Il s'agit d'une extraction d'un masque binaire basé profondeur, suivi du codage de texture basé RoI. Ce qui diffère ici, c'est que 1) l'extraction du masque ne se fait pas à partir de la représentation en régions, mais à partir de la moyenne de la profondeur reconstruite par région du résultat de segmentation, et 2) le raffinement s'applique uniquement sur la texture.

6.3.3.1 *Extraction du Masque-DoI*

Cette extraction est réalisée en deux étapes :

1) calculer la moyenne de la profondeur par région du résultat de segmentation, en utilisant les valeurs de profondeur de la carte reconstruite, pour avoir une carte $\mathrm{Prof}_{\mathrm{Moy/Reg}}$,
2) binariser la carte $\mathrm{Prof}_{\mathrm{Moy/Reg}}$, avec l'intervalle de profondeur $[Z_{\mathrm{in\_l}}, Z_{\mathrm{in\_h}}]$ comme paramètre d'entrée (voir FIGURE 125). La carte de profondeur considérée est celle qui est reconstruite, et ceci pour deux raisons principales. La première est que le processus doit être dupliqué au décodeur. La deuxième est que le masque doit être vu comme une partie de la partition QuadTree.

La FIGURE 126 donne un exemple d'extraction du masque binaire (Masque-DoI) de Newspaper (1024x768 pixels).

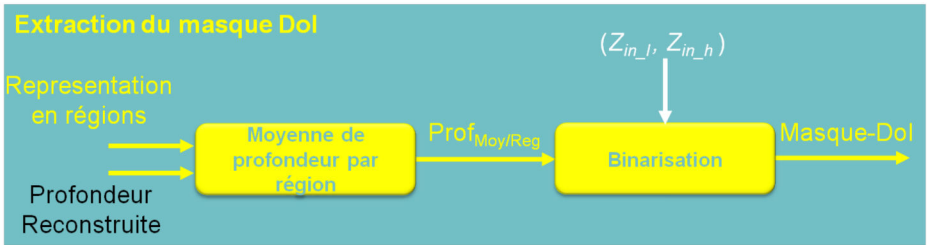
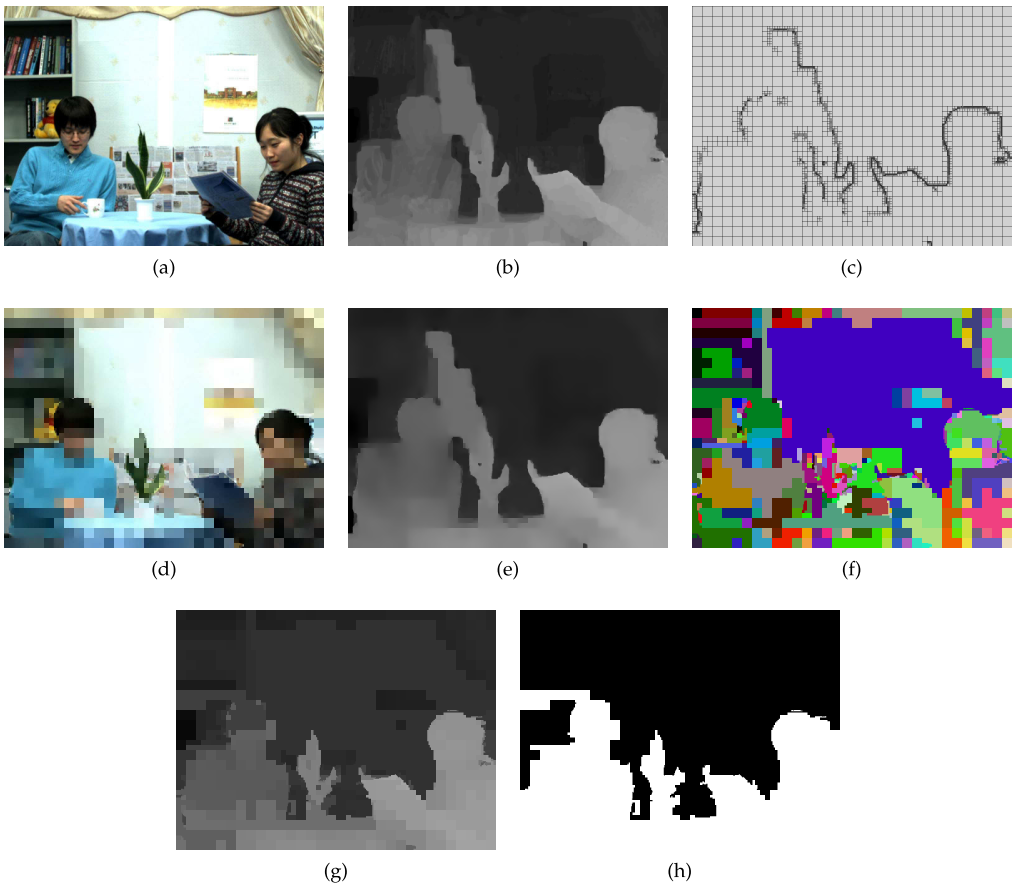


FIGURE 125. Représentation de la DoI par extraction du Masque-DoI.

6.3.3.2 *Raffinement de la DoI dans la texture*

La procédure de raffinement de qualité de la DoI dans la texture est identique à celle présentée dans la Section 6.2.5. Ce raffinement est exclusif aux objets se localisant dans la zone de profondeur d'intérêt. Ainsi, les deux approches proposées dans la section 6.2.5 sont applicables : i) raffiner le SNR de la DoI utilisant le concept de codage par Région d'Intérêt (RoI), ii) raffiner la résolution locale de la texture à l'intérieur de la DoI uniquement, en utilisant la grille $\mathrm{Quad}_{\mathrm{ZT\_DoI}}$. Une solution jointe de raffinement de SNR et de résolution locale de la DoI reste bien évidemment réalisable.



(a) (b) (c)

(d) (e) (f)

(g) (h)

FIGURE 126. Exemple d'extraction du masque binaire (Masque-DoI) de Newspaper (1024x768 pixels) vue 6 image 1 : (a) texture originale ; (b) profondeur originale ; (c) grille profondeur (Th_{Quad} = 46) ; (d) texture basse résolution (0.042 bpp, PSNR = 18,26 db) ; (e) profondeur reconstruite (0.031 bpp, PSNR = 36,75 db) ; (f) segmentation basée luminance (Th_{Cost} = 6, 272 régions) ; (g) moyenne de la profondeur par région ($Prof_{Moy/Reg}$) ; (h) Masque-DoI avec Z_{in_l} = 63, Z_{in_h} = 255.

6.3.4 *Expérimentation et résultats d'Autofocus 3D avancé basé segmentation*

Les tests sont effectués sur les images 3D de référence fournies par MPEG. Nous appliquons la segmentation sémantique suivie de l'Autofocus 3D. Les paramètres à régler sont :

- $(Q_{p\_T\_BR}, Th_{Quad})$ pour la compression à basse résolution,
- $(\alpha, \beta_1, \beta_2, \gamma)$ pour le choix de la sémantique entre 2D et Z,
- Th_{Cost} pour le critère de fusion de la segmentation,
- Z_{in_l}, Z_{in_h} et F pour l'Autofocus,
- $Q_{p\_T\_Raf\_DoI}$ et $Q_{p\_T\_Raf\_NDoI}$ pour le raffinement.

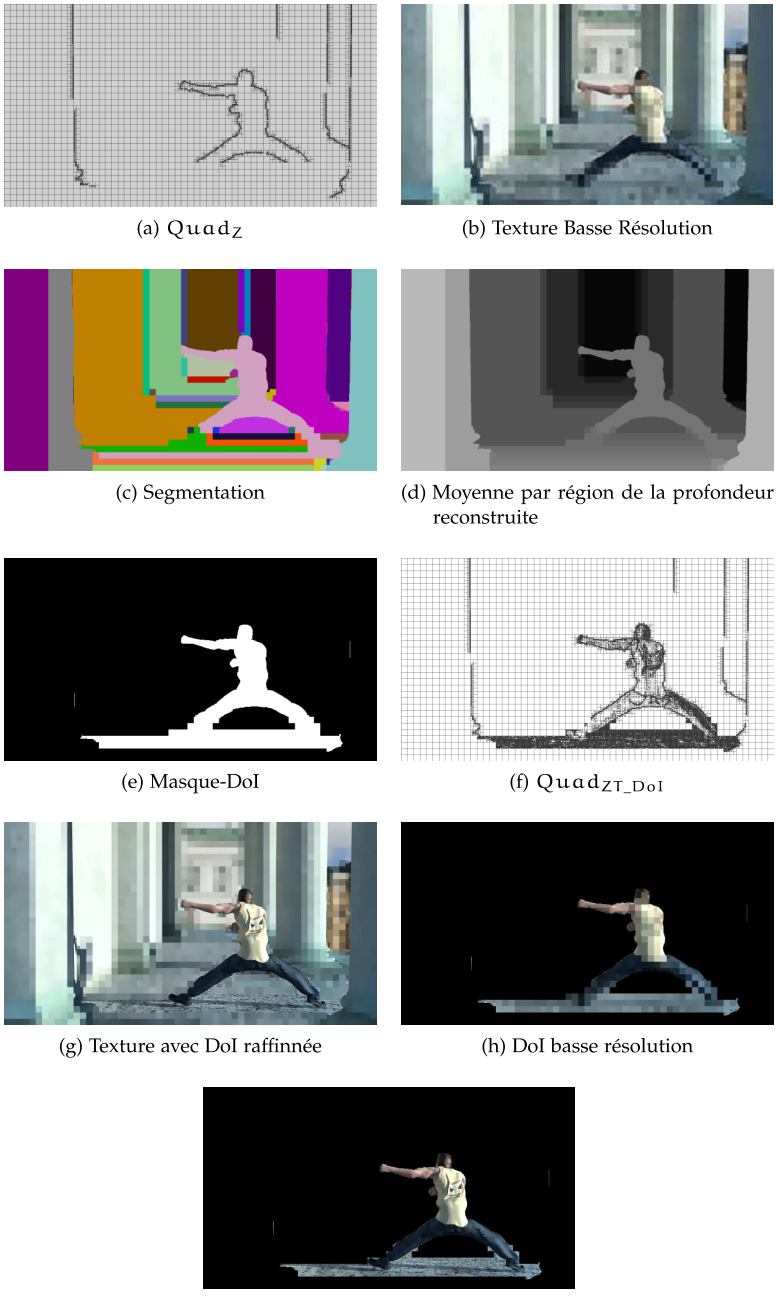
Dans ces exemples nous considérons le cas de raffinement de résolution locale uniquement. Nous prenons ainsi $Q_{p\_T\_BR} = Q_{p\_T\_Raf\_DoI} = Q_{p\_T\_Raf\_DoI}$.

La comparaison avec les techniques de segmentation existantes n'est pas réalisable. En effet, au mieux de nos connaissances, l'algorithme global de représentation et de codage proposé est unique en termes de fonctionnalités combinées.

Les FIGURES 127 et 128 montrent des exemples de résultats incluant la grille de profondeur, la texture à basse résolution, la représentation en région, la moyenne par région de la profondeur reconstruite, le Masque-DoI, la grille raffinée, la texture raffinée, la RoI extraite de la texture à basse résolution, et la RoI extraite de la texture raffinée de *Undodancer* et *Balloons*.

Il est bien clair dans les deux exemples donnés, que les RoI sont extraites avec une grande précision et le raffinement est exclusif aux objets d'intérêt. Ceci assure une cohérence entre la texture, la profondeur et les régions tout en minimisant les problèmes de distorsions au niveau des contours.

Ainsi, les résultats des deux solutions sont d'assurer une haute consistance entre la texture, la profondeur et les régions.



(a) Quad_Z (b) Texture Basse Résolution

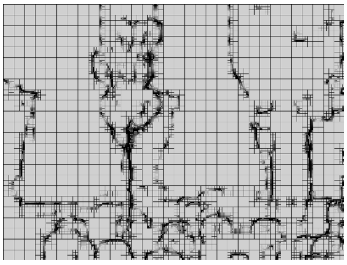
(c) Segmentation (d) Moyenne par région de la profondeur reconstruite

(e) Masque-DoI (f) Quad_{ZT_DoI}

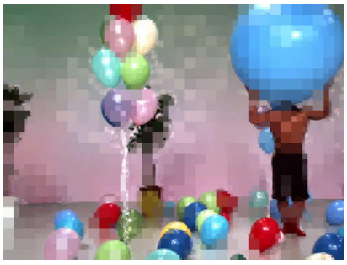
(g) Texture avec DoI raffinnée (h) DoI basse résolution

(i) DoI raffinée

FIGURE 127. Résultats de raffinement de UndoDancer vue 1 image 250 : (a) Quad_Z ($\mathrm{Th}_{\mathrm{Quad}} = 33$) à 0.003 bpp ; (b) Texture Basse Résolution reconstruite à 0.04 bpp ($Q_p = 50$, PSNR = 17dB) ; (c) une segmentation basée profondeur ($\mathrm{Th}_{\mathrm{Cost}} = 2$) ; (d) moyenne par région de la profondeur reconstruite ; (e) Masque-DoI avec $Z_{in\_l} = 110$, $Z_{in\_h} = 121$; (f) Grille raffinée à 0.03 bpp ; (g) Texture raffinée à 0.8 bpp, (PSNR = 22dB) ; (h) RoI extraite à basse résolution (PSNR = 12.09dB) ; (i) RoI extraite à haute résolution (PSNR = 30dB).



(a) Quad_Z

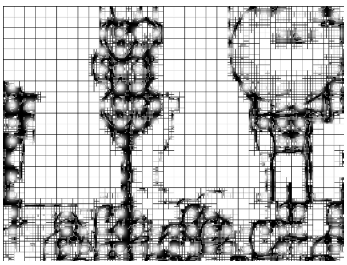


(b) Texture Basse Résolution

(c) Segmentation

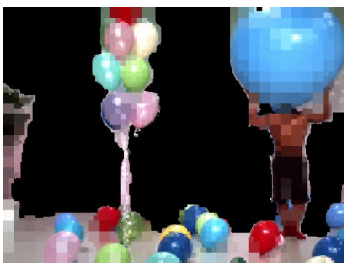
(d) Moyenne par région de la profondeur reconstruite

(e) Masque-DoI

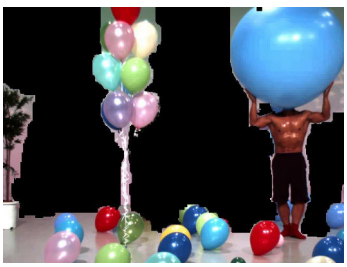


(f) Quad_{ZT_DoI}

(g) Texture avec DoI raffinnée



(h) DoI basse résolution



(i) DoI raffinée

FIGURE 128. Résultats de raffinement de Balloons vue 5 image 1 : (a) Quad_Z ($\mathrm{Th}_{\mathrm{Quad}} = 16$) à 0.013 bpp ; (b) Texture Basse Résolution reconstruite à 0.14 bpp ($Q_p = 25$, PSNR = 22.61dB) ; (c) une segmentation basée profondeur ($\mathrm{Th}_{\mathrm{Cost}=2}$) ; (d) moyenne par région de la profondeur reconstruite ; (e) Masque-DoI avec $Z_{in\_l} = 128$, $Z_{in\_h} = 255$; (f) Grille raffinée à 0.03 bpp ; (g) Texture raffinée à 2 bpp, (PSNR = 27dB) ; (h) RoI extraite à basse résolution (PSNR = 16.51dB) ; (i) RoI extraite à haute résolution (PSNR = 33.3dB).

Dans ce chapitre nous avons proposé deux solutions toutes deux basées sur le schéma scalable 2D+Z de compression du Chapitre 4, pour un schéma global et complet de représentation "pixelique" et codage niveau région dédié aux images 3D.

Dans la première partie du chapitre, nous avons introduit un schéma joint de représentation et de codage de Profondeur d'Intérêt DoI, appelé "Autofocus 3D simple". Il vise à se focaliser sur une zone de profondeur d'intérêt DoI dans la scène. La représentation de la DoI consiste ainsi en une extraction fine d'un masque de la carte de profondeur, avec une résolution pixelique. Le codage DoI concerne ensuite à la fois la profondeur et la texture. La première étape est un pré-traitement appliqué à la carte de profondeur, qui a pour but d'ajuster la dynamique afin d'augmenter la résolution de la grille du QuadTree au niveau de la DoI. La texture est ensuite compressée en utilisant un schéma de codage basé RoI, dont le masque est déduit de la carte de profondeur. Ce procédé permet un raffinement de qualité et/ou de la résolution locale de la texture au niveau de la zone de profondeur d'intérêt uniquement. La technique d'Autofocus 3D assure une haute qualité de reconstruction sur les contours des objets dans les images de texture et de profondeur. Cette qualité est maintenue dans les vues synthétisées, de part la forte corrélation du codage entres les différentes composantes.

Nous avons ensuite proposé dans la deuxième partie du chapitre un schéma "Autofocus 3D avancé" dédié à la représentation en régions et à l'extraction/raffinement de la RoI. Dans ce schéma, une segmentation, guidée par la sémantique 3D, est appliquée sur la version basse résolution de l'image 3D. La segmentation 3D proposée préserve les contours principaux des objets de la scène avec une résolution pixelique. Finalement, un raffinement local est réalisé pour la RoI déduite de la représentation en régions. Les résultats ont montré que la méthode peut extraire finement à différents niveaux de granularité le contenu de la scène. Par ailleurs, un raffinement précisément localisé aux contours de la RoI peut être effectué, à n'importe quel niveau de qualité. Comme dans le cas de l'Autofocus 3D simple, l'élément le plus remarquable de l'approche est d'assurer une pleine cohérence spatiale entre texture, profondeur, et régions, se traduisant par une meilleure qualité dans les vues synthétisées.

7 CONCLUSION

La technologie 3D et ses applications impliquent des contraintes incontournables lors de la conception de schémas de codage d'images 3D. La scalabilité est ainsi une contrainte essentielle pour tenter de répondre à l'hétérogénéité de qualité des différentes applications. En outre, la capacité d'extraire la sémantique est indispensable pour pouvoir interpréter et donc mieux exploiter ces images.

À cette fin, plusieurs questions pourront être posées : comment combler la manque de cohérence entre la texture et la profondeur de la scène ? comment coupler une représentation fine des contours avec un schéma de codage qui peut être lui basé contenu d'images ?

Dans un contexte applicatif tel que la plateforme 3D introduite dans le Chapitre 2, et en considérant l'ensemble des contraintes associées, nous avons proposé un schéma de codage 2D+Z scalable et joint texture/profondeur avec des fonctionnalités avancées. Notre contribution se base sur le codec LAR, initialement conçu pour les images 2D. Ce schéma est composé de deux étapes : un schéma de compression à basse résolution et un schéma de compression à haute résolution.

Le premier consiste à fournir une version à bas débits de l'image 3D, tout en préservant les contours. La texture est tout d'abord codée suivant la grille du Quadtree basée profondeur. Ceci fournit une texture à basse résolution tout en maintenant la consistance entre la profondeur et la texture. Ensuite, la profondeur est codée conjointement avec la texture à basse résolution, avec une approche LARP (LAR pour Profondeur) qui consiste à

1) coder la profondeur en exploitant sa forte corrélation avec la texture : prédire la profondeur avec le meilleur prédicteur de la texture sélectionné a posteriori,

2) éliminer d'une manière homogène les effets blocs de la carte de profondeur reconstruite à l'aide d'une interpolation adaptative.

Les résultats ont montré que l'approche LARP proposée dans le schéma de codage à basse résolution permet de réduire le débit de codage de la carte de profondeur et d'améliorer la qualité visuelle à la fois des cartes de profondeur reconstruites et des vues synthétisées, notamment sur les contours des objets.

Le schéma de compression à haute résolution quant à lui permet un rehaussement de qualité de la texture afin de fournir une image 3D à plus haute qualité. Il consiste en un raffinement de la grille simple de la profondeur en utilisant la texture, suivie par le raffinement de la texture. Les expérimentations ont montré que le schéma proposé assure une scalabilité pour un léger surcoût par rapport au schéma de codage non scalable.

Dans le Chapitre 6, un schéma global de représentation fine et de codage niveau région et dédié aux images 3D est proposé, afin d'unifier les notions de forme et de contenu.

Nous avons introduit dans la première partie du chapitre un schéma joint de représentation et codage de Profondeur d'Intérêt DoI, appelé "Autofocus 3D simple". Ce schéma vise à se focaliser sur une zone de Profondeur d'Intérêt (DoI) dans la scène. Il s'agit d'une nouvelle approche qui consiste en :

1) un ajustement de la dynamique de la carte de profondeur originale afin d'incrémenter le niveau d'activité à l'intérieur de la DoI. Ceci assure une résolution plus fine pour la DoI dans la grille de profondeur et par suite à la carte reconstruite.

2) une extraction fine des objets, suivie d'un raffinement local à l'intérieur de la DoI de qualité SNR et/ou de résolution.

Ensuite, nous avons proposé dans la deuxième partie du chapitre, un schéma "Autofocus 3D avancé" dédié à la représentation en régions et à l'extraction/raffinement de la RoI. Dans ce schéma, une segmentation, guidée par la sémantique 3D, est proposée. Un contrôle du degré de granularité est offert grâce à plusieurs paramètres : coefficients de pondération pour les différentes composantes et seuil de segmentation adaptatif. À l'issue de cette segmentation, une

représentation interprétable de la scène est fournie. Une extraction fine et un raffinement exclusif à la DoI, suivant le même principe de l'Autofocus 3D simple, sont ensuite appliqués.

L'élément le plus remarquable de ces deux approches est d'assurer une pleine cohérence spatiale entre texture, profondeur, et régions, se traduisant par une minimisation des problèmes de distorsions au niveau des contours et ainsi par une meilleure qualité dans les vues synthétisées.

PERSPECTIVES

Le codage de la profondeur proposé dans la première partie de cette thèse, appelé LARP (LAR pour Profondeur), peut prétendre à plusieurs améliorations. En particulier, la technique du meilleur prédicteur simple proposée peut être plus sophistiquée. Nous pouvons ainsi ajouter plus de prédicteurs lors du choix de la meilleure prédiction a posteriori de la texture. Par exemple, il est possible de chercher de nouvelles directions pour la prédiction. En contrepartie, ceci se fera au détriment de la complexité du code et du temps de calcul.

En outre, le seuil de l'interpolation adaptative proposée est sélectionné manuellement pour chaque image 3D. Une étude plus approfondie peut être réalisée pour une sélection automatique du seuil d'interpolation, avec par exemple un seuil variable dans l'image. Le seuil peut être ainsi rendu adaptatif en fonction du gradient local de la carte de profondeur.

D'autre part, pour l'Autofocus 3D, les paramètres Z_{in_l} et Z_{in_h} de la carte de profondeur sont sélectionnés manuellement en fonction de l'objet d'intérêt de la scène. Il est possible de chercher une solution automatique pour déterminer les limites de la profondeur d'intérêt. Une piste envisageable consiste à coupler la segmentation proposée à un algorithme de détection de contours, suivi d'une estimation des objets saillants dans la scène. Les paramètres Z_{in_l} et Z_{in_h} peuvent être ensuite choisis en fonction de la carte de profondeur en fonction du résultat d'objets saillants.

Une perspective directe d'un tel procédé consiste à étudier le cas où la scène présente plusieurs profondeurs d'intérêt : la focalisation n'est plus sur un seul intervalle de profondeur, mais sur plusieurs. Nous parlons ainsi d'un "Multifocus". Ceci engendrera une réduction de la qualité maximale possible pour une seule région d'intérêt avec un débit constant. Une évaluation de la qualité des vues intermédiaires synthétisées est ainsi nécessaire afin de juger l'efficacité d'un tel schéma de "Multifocus".

Par ailleurs, un autre axe important est celui de l'étude de la complexité des schémas proposés, suivie d'une optimisation du code. Ces travaux ont en réalité été partiellement réalisés, sur la partie LARP classique, dans le cadre d'un stage master. L'utilisation d'outils tels que VTune, qui aident à mesurer le temps d'exécution des différentes fonctions et déterminer la quantité de mémoire utilisée par chaque fonction lors de son appel, ont permis une optimisation partielle. D'autres gains sur la complexité peuvent notamment être obtenus pour les algorithmes de segmentation.

Enfin, les schémas proposés, appliqués aux images fixes, peuvent avoir une extension à la vidéo 3D. Les différentes techniques proposées exploitent la corrélation spatiale qui existe dans l'image 3D elle même. Dans l'extension 3D, de telles techniques peuvent en plus exploiter la corrélation temporelle entre les images de la séquence vidéo. En particulier, la technique de meilleur prédicteur peut utiliser des prédicteurs temporels à partir des images temporellement voisins. Pour l'Autofocus 3D, l'adaptation doit tenir compte de problèmes liés au suivi des objets d'intérêt tels que les déformations, les mouvements de translation complexes, ou encore des occlusions partielles ou totales.

TABLE DES FIGURES

LISTE DES TABLEAUX

LISTE DES ABRÉVIATIONS

2D Deux Dimensions

2D+Z 2D plus profondeur

3D Trois Dimensions

3DTV Télévision Trois Dimensions

AVC Advanced Video Coding

B Bi-direcional predictive frame

BC Backward Compatibility

BP best predictor

CFP Cal for Proposal

DCP Disparity Compensation Prediction

DCT Transformation par Cosinus Discrète

DIBR Depth Image Based Rendering

DMM Depth Map Modeling

DOI Depth of Interest

DV Disparity Vector

FC Forward Compatibility

FC frame compatible

FTV Free Viewpoint TV

FVV Free Viewpoint Video

GMM Gaussian Mixture Model

GOP Group of Pictures

HEVC High Efficient Video Coding

I Intra coded frame

JVDF Joint View Depth Filtering

JVT Joint Video Team

LAR Locally Adaptive Resolution

LARP LAR pour Profondeur

MCP Motion Compensation Prediction

MPEG Moving Picture Expert Group

MPI Motion Prediction Inheritance

MV Motion Vector

MVC Multiview Video Coding

MVC+D Multiview Video plus Depth coding

MVD Multiview Video plus Depth

MVP MultiView Profile

MVV MultiView Video

P Predictive frame

PSNR Peak Signal to Noise Ratio

ROI Région d'Intérêt

SNR Signal to Noise Ratio

SSE Sum Squared Errors

SVDC Synthesised View Distortion Change

SVH Système Visuel Humain

TOF Time of Flight

VCEG Video Coding Expert Group

VLC Variable Length Code

VSP View Synthesis Prediction

BIBLIOGRAPHIE

[1] M. Pollefeys, R. Koch, , and L. Van Gool, "A simple and efficient rectification method for general motion," *Proceedings of the Seventh IEEE International Conference on computer Vision*, vol. 1, pp. 496–501, Sept. 1999.

[2] A. Fusiello, E. Trucco, and A. Verri, "A compact algorithm for rectification of stereo pairs," *Machine Vision Application*, vol. 12, no. 1, pp. 16–22, Mar. 2000.

[3] S. Gurbuz, M. Kawakita, and H. Ando, "Color calibration for multi-camera imaging systems," Oct 2010, pp. 201–206.

[4] Youngbae Hwang, Je Woo Kim, Byeong-Ho Choi, and Wangheon Lee, "Color correction without color patterns for stereoscopic camera systems," *11th International Conference on Control, Automation and Systems (ICCAS)*, pp. 1129–1134, Oct. 2011.

[5] W. Kim, J. Kim, Minsu Choi, Ik-Joon Chang, and Jinsang Kim, "Low complexity image correction using color and focus matching for stereo video coding," *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 2912–2915, May 2013.

[6] P. Kauff, N. Atzpadin, C. Fehn, M. Muller, O. Schreer, A. Smolic, and R. Tanger, "Depth map creation and image-based rendering for advanced 3dtv services providing interoperability and scalability," *Signal Processing :Image Communication. Special Issue on 3DTV*, Feb. 2007.

[7] O. Stankiewicz, K. Wegner, and M. Domanski, "Nonlinear depth representation for 3d video coding," *IEEE International Conference on Image Processing (ICIP)*, pp. 1752–1756, Sept. 2013.

[8] H. Fradi and J.L Dugelay, "Improved depth map estimation in stereo vision," *Electronic Imaging Conference on 3D Image Processing and Applications 3DIP*, vol. 7863, Jan. 2011.

[9] S. Foix, G. Alenya, and C. Torras, "Lock-in time-of-flight (tof) cameras : A survey," *IEEE Sensors Journal*, vol. 11, no. 9, pp. 1917–1926, Sept. 2011.

[10] I. Schiller, B. Bartczak, F. Kellner, and Reinhard Koch, "Increasing realism and supporting content planning for dynamic scenes in a mixed reality system incorporating a time-of-flight camera," *Journal of Virtual Reality and Broadcasting*, vol. 7, no. 4, 2010.

[11] J. Park, H. Kim, Y.W. Tai, M.S. Brown, and I. Kweon, "High quality depth map upsampling for 3d-tof cameras," *IEEE International Conference on Computer Vision (ICCV)*, pp. 1623–1630, Nov. 2011.

[12] P. Rander, P. J. Narayanan, and T. Kanade, "Virtualized reality : constructing time-varying virtual worlds from real world events," *Proceedings of the IEEE Visualization*, pp. 277–283, Oct 1997.

[13] ITU-T, "Advanced video coding for generic audiovisual services," *Rapport, ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 AVC)*, 2010.

[14] B. Brossa, W. J. Han, J. R. Ohm, G. Sullivan, and T. Wiegand, "High efiiciency video coding (hevc) text specification draft 9," *Rapport, ITU-T SG16 WP3 & ISO/IEC JTC1/SC29/WG11 JCTVC-K1003*, Oct. 2012.

[15] J. R. Ohm and G. Sullivan, "High efficiency video coding : The next frontier in video compression," *IEEE Signal Processing Magazine*, vol. 30, no. 1, 2013.

[16] J.R. Ohm, "Overview of 3d video coding standards," *Proceedings of 3DSA*, 2013.

[17] J. Lim, K. Ngan, W. Yang, and K. Sohn, "A multiview sequence codec with view scalability," *Sinal Processing : Image Communication*, vol. 19, no. 3, pp. 239–256, 2004.

[18] S. Shimizu, M. Kitahara, H. Kimata, K. Kamikura, and Y. Yashima, "View scalable multiview video coding using 3-d warping with depth map," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 11, pp. 1485–1495, Nov. 2007.

[19] A. Vetro, "Frame compatible formats for 3d video distribution," *IEEE International Conference on Image Processing (ICIP)*, pp. 2405–2408, Sept. 2010.

[20] L. Lucas, C. Loscos, and Y. Remion, "Vidéo 3d : capture, traitement et diffusion," *Lavoisier*, 2013.

[21] H. Shum and S.B. Kang, "Review of image-based rendering techniques," *Proceedings of SPIE, Visual Communications and Image Processing*, vol. 2, May 2000.

[22] S.C. Chan, H.Y. Shum, and K.T. Ng, "Image-based rendering and synthesis," *Signal Processing Magazine, IEEE*, vol. 24, no. 6, pp. 22–33, Nov 2007.

[23] P. Ndjiki-Nya, M. Koppel, D. Doshkov, H. Lakshman, P. Merkle, K. Muller, and T. Wiegand, "Depth image-based rendering with advanced texture synthesis for 3-d video," *IEEE Transactions on Multimedia*, vol. 13, no. 3, pp. 453–465, Jun. 2011.

[24] K. Muller, A. Smolic, K. Dix, P. Merkle, P. Kauff, and T. Wiegand, "View synthesis for advanced 3d video systems," *Eurasip Journal on Image and Video Processing*, vol. 2008, 2008.

[25] Q.H. Nguyen, M.N. Do, and S.J. Patel, "Depth image-based rendering with low resolution depth," *IEEE International Conference on Image Processing (ICIP)*, pp. 553–556, Nov. 2009.

[26] M. Tanimoto, T. Fuji, K. uzuki, N. Fukushima, and Y. Mori, "Reference softwares for depth estimation and view synthesis," *ISO/IEC JTC1/SC29/WG11MPEG2008/M15377*, Apr. 2008.

[27] J. Gautier, O. Le Meur, and C. Guillemot, "Depth-based image completion for view synthesis," *3DTV Conference : The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, May 2011.

[28] E. Bosc, R. R. Pepion, P. Le Callet, M. Koppel, P. Ndjiki-Nya, M. Pressigout, and L. Morin, "Towards a new quality metric for 3-d synthesized view assessment," *IEEE Journal on Selected Topics in Signal Processing*, vol. 5, no. 7, pp. 1332–1343, Nov. 2011.

[29] E. Peinsipp-Byma, N. Rehfeld, and R. Eck, "Evaluation of stereoscopic 3d displays for image analysis tasks," *Proceeding od SPIE, Stereoscopic Display and Aplications*, Jan. 2009.

[30] Y. Zhu and T. Zhen, "3d multi-view autostereoscopic display and its key technologie," *Asia-Pacific Conference on Information Processing (APCIP)*, vol. 2, pp. 31–35, Jul. 2009.

[31] B. Mercier, K. Boulanger, C. Bouville, and K. Bouatouch, "Multiview autostreoscopic displays," *Publiction interne N°1868, inria-00192688, version 1*, Nov. 2007.

[32] T. Peterka, R.L. Kooima, J.I. Girado, G. Jinghua, D.J. Sandin, A. Johnson, J. Leigh, J. Schulze, and T.A. DeFanti, "Dynallax : Solid state dynamic parallax barrier autostereoscopic vr display," pp. 155–162, Mar. 2007.

[33] T. Colleu, "A floating polygon soup representation for 3d video," *http :\\tel.archives-ouvertes.fr\docs\00\59\22\07\PDF\thA_se_colleu\degre_prepress.pdf*, 2010.

[34] D. Jung and R. Koch, "Efficient depth-compensated interpolation for full parallax displays," *Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*, May 2010.

[35] G.E. Favalora, "Volumetric 3d displays and application infrastructure," *Computer*, vol. 38, no. 8, 2005.

[36] C. Slinger, C. Cameron, and M. Stanley, "Computer-generated holography as a generic display technology," *Computer*, vol. 38, no. 8, pp. 46–53, Aug. 2005.

[37] J. Jiang, M.G. Liu, and C.H. Hou, "Texture-based image indexing in the process of lossless data compression," *IEEE Proceedings on Vision, Image and Signal Processing*, vol. 150, no. 3, pp. 198–204, Jun. 2003.

[38] L. Lei, J. Peng, and B. Yang, "72-trees index for image retrieval," *IEEE International Conference on Cognitive Informatics Cognitive Computing (ICCI*CC)*, pp. 268–273, Aug. 2012.

[39] W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos, and G. Taubin, ," .

[40] A. Pentland, R.W. Picard, and S. Sclaroff, ," .

[41] M. L. Kherfi, D. Ziou, and A. Bernardi, "Image retrieval from the world wide web : Issues, techniques, and systems," *ACM Computer Survey*, vol. 36, no. 1, pp. 35–67, Mar. 2004.

[42] Y.A. ASLANDOGAN and C.T. YU, "Diogenes : A web search agent for content based indexing of personal images," *ACM Computer Survey*, 2000.

[43] M. L. Kherfi, D. Ziou, and A. Bernardi, "Wise : A web-based image retrieval engine," *Proceedings of the International Conference on Image and Signal Processing (ICISP)*, pp. 69–77, 2003.

[44] E.Y. CHANG, J. WANG, C. LI, and G. WEIDERHOLD, "Rime : A replicated image detector for the world-wide web," *Proceedings of the SPIE Symposium of Voice, Video, and Data Communications*, pp. 58–67, 1998.

[45] K. BARNARD, P. DUYGUL, and D.A. FORSYTH, "Clustering art," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, p. 434–441, 2001.

[46] C.C. CHEN and J.Z. WANG, "Digital library and semantics-sensitive region-based retrieval," *Proceedings of the International Conference on Digital Library—IT Opportunities and Challenges in the New Millennium*, p. 454–462, 2002.

[47] K. Muller, P. Merkle, and T. Wiegand, "3-d video representation using depth maps," *Proceedings of the IEEE*, vol. 99, no. 4, pp. 643–656, Apr. 2011.

[48] P. Merkle, K. Muller, A. Smolic, and T. Wiegand, "Efficient compression of multi-view video exploiting inter-view dependencies based on h.264/mpeg4-avc," *IEEE International Conference on Multimedia and Expo*, pp. 1717–1720, Jul. 2006.

[49] T.Y. Kuo, C.K. Yeh, and H.Y. Tsai, "A novel method for global disparity vector estimation in multiview video coding," *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 864–867, May 2009.

[50] S.H. Lee, S.H. Lee, N.k. Cho, and J.H. Yang, "A motion vector prediction method for multiview video coding," *International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP)*, pp. 1247–1250, Aug. 2008.

[51] C.H. Chen, S.C. Lee, and J.J. Chen, "An improved block matching and prediction algorithm for multi-view video with distributed video codec," *IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, Jul. 2011.

[52] K.M. Wong, L.M. Po, K.W. Cheung, C.W. Ting, K.H. Ng, and X. Xu, "Horizontal scaling and shearing-based disparity-compensated prediction for stereo video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 10, pp. 1457–1470, Oct. 2012.

[53] W. Zhu, P. Chen, Y. Zheng, and J. Feng, "Fast inter-view prediction and mode selection for multiview video coding," *International Congress on Image and Signal Processing (CISP)*, vol. 1, pp. 151–156, Oct 2011.

[54] W. Su, D. Rusanovskyy, M.M. Hannuksela, and H. Li, "Depth-based motion vector prediction in 3d video coding," *Proceeding of Picture Coding Symposium*, pp. 37–40, May 2012.

[55] R. Li, D. Rusanovskyy, M.M. Hannuksela, and H. Li, "Joint view filtering for multiview depth map sequences," *Proceeding of IEEE International Conference Image Processing*, pp. 1329–1332, Sept. 2012.

[56] M.M Hannuksela, D. Rusanovskyy, W. Su, L. Chen, R. Li, P. Aflaki, L. Deyan, M. Joachimiak, L. Houqiang, and M. Gabbouj, "Multiview-video-plus-depth coding based on the advanced video coding standard," *IEEE Transactions on Image Processing*, vol. 22, no. 9, pp. 3449–3458, Sept. 2013.

[57] K. Muller, H. Schwarz, D. Marpe, C. Bartnik, S. Bosse, H. Brust, T. Hinz, H. Lakshman, P. Merkle, F.H. Rhee, G. Tech, M. Winken, and T. Wiegand, "3d high-efficiency video coding for multi-view video and depth data," *IEEE Transactions on Image Processing*, vol. 22, no. 9, pp. 3366–3378, Sept. 2013.

[58] P. Aflaki, M. Hannuksela, D. Rusanovskyy, and M. Gabbouj, "Nonlinear depth map resampling for depth-enhanced 3-d video coding," *IEEE Signal Processing Letters*, vol. 20, no. 1, pp. 87–90, Jan. 2013.

[59] K.J. Oh, S. Yea, A. Vetro, and Y.S. Ho, "Depth reconstruction filter and down/up sampling for depth coding in 3-d video," *IEEE Signal Processing Letters*, vol. 16, no. 9, pp. 747–750, Sept. 2009.

[60] S. Yea and A. Vetro, "View synthesis prediction for multiview video coding," .

[61] P. Lai, D. Tian, and P. Lopez, "Depth map processing with iterative joint multilateral filtering," pp. 9–12, Dec. 2010.

[62] E. Ekmekcioglu, V. Velisavljevic, and S.T. Worrall, "Content adaptive enhancement of multiview depth maps for free viewpoint video," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 2, pp. 352–361, Apr. 2011.

[63] D. Farin Y. Morvan and P. H. N. de With, "Platelet-based coding of depth maps for the transmission of multiview images," *Proceedings of SPIE*, vol. 6055, Jan. 2006.

[64] B. Kamolrat, W.A.C. Fernando, and M. Mrak, "3d motion estimation for depth information compression in 3d-tv applications," *IET Electronics Letters*, vol. 44, no. 21, pp. 1244–1245, Oct. 2008.

[65] B. Kamolrat, W.A.C Fernando, and M. Mrak, "Adaptive motion-estimation-mode selection for depth video coding," *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pp. 702–705, Mar. 2010.

[66] J. Gautier, O. Le Meur, and C. Guillemot, "Efficient depth map compression based on lossless edge coding and diffusion," *Picture Coding Symposium (PCS)*, pp. 7–9, May 2012.

[67] I. Daribo, G. Cheung, and D. Florencio, "Arithmetic edge coding for arbitrarily shaped sub-block motion prediction in depth video compression," *IEEE International Conference on Image Processing (ICIP)*, pp. 1541–1544, Sept. 2012.

[68] J.Y. Lee, H.C. Wey, and D.S. Park, "A fast and efficient multi-view depth image coding method based on temporal and inter-view correlations of texture images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 12, pp. 1859–1868, Dec. 2011.

[69] J. Seo, D.Park, H.C. Wey, S. Lee, and K. Sohn, "Motion information sharing mode for depth video coding," *Proceedings 3DTV Conference*, pp. 1–4, June 2010.

[70] M. Winken, H. Schwarz, and T. Wiegand, "Motion vector inheritance for high efficiency 3d video plus depth coding," *Picture Coding Symposium (PCS)*, pp. 53–56, May 2012.

[71] S. Milani, P. Zanuttigh, M. Zamarin, and S. Forchhammer, "Efficient depth map compression exploiting segmented color data," *IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, Jul. 2011.

[72] I. Daribo, C. Tillier, and B. Pesquet-Popescu, "Adaptive wavelet coding of the depth map for stereoscopic view synthesis," *IEEE Workshop on Multimedia Signal Processing*, pp. 413–417, Oct. 2008.

[73] P. Merkle, Y. Morvan, A. Smolic, D. Farin, K. Muller, P.H.N. de With, and T. Wiegand, "The effect of depth compression on multiview rendering quality," *3DTV Conference : The True Vision - Capture, Transmission and Display of 3D Video, 2008*, pp. 245–248, May 2008.

[74] A. Tikanmaki, A. Gotchev, A. Smolic, and K. Miller, "Quality assessment of 3d video in rate allocation experiments," *IEEE International Symposium on Consumer Electronics, (ISCE)*, pp. 1–4, Apr. 2008.

[75] E. Bosc, V. Jantet, M. Pressigout, L. Morin, and C. Guillemot, "Bit-rate allocation for multi-view video plus depth," *3DTV Conference : The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, pp. 1–4, May 2011.

[76] W.S. Kim, A. Ortega, P. Lai, D. Tian, and C. Gomila, "Depth map distortion analysis for view rendering and depth coding," *IEEE International Conference on Image Processing (ICIP)*, pp. 721–724, Nov. 2009.

[77] O. Deforges, M. Babel, L. Bedat, and J. Ronsin, "Color lar codec : A color image representation and compression scheme based on local resolution adjustment and self-extracting region representation," *TCSVT*, vol. 17, pp. 974–987, 2007.

[78] O. Deforges, "Codage d'images par la méthode lar et méthodologie, adéquation algorithme architecture. de la définition des algorithmes de compression au prototypage rapide sur architectures parallèles hétérogènes," Nov. 2004.

[79] K. Samrouth, O. Deforges, and F. Pasteau, "Quality constraint and rate-distortion optimization for predictive image coders," *Proc. SPIE - Image Processing : Algorithms and Systems XI*, vol. 8655, pp. 1–9, Feb. 2013.

[80] M. Babel and O. Deforges, "Lossless and lossy minimal redundancy pyramidal decomposition for scalable image compression technique," *IEEE ICASSP*, vol. 3, pp. 249–252, 2003.

[81] Y. Liu, O. Deforges, F. Pasteau, and K. Samrouth, "One pass quality control and low complexity rdo in a quadtree based scalable image coder," *IEEE International Conference on Image Information Processing (ICIIP)*, pp. 187–192, Dec. 2013.

[82] F. Pasteau, C. Strauss, M. Babel, O. Deforges, and L. Bedat, "Improved colour decorrelation for lossless colour image compression using the lar codec," *European Signal Processing Conference (EUSIPCO)*, 2009.

[83] F. Pasteau, C. Strauss, M. Babel, O. Deforges, and L. Bedat, "Adaptive color decorrelation for predictive image codecs," *European Signal Processing Conference (EUSIPCO)*, 2011.

[84] R. Awad, K. Samrouth, and W. Falou, "Implementation de la méthode de compression lar dans la raspberry-pi," 2014.

[85] P.H Batavia and S. Singh, "Obstacle detection using adaptive color segmentation and color stereo homography," *IEEE International Conference on Robotics and Automation*, vol. 1, pp. 705–710, 2001.

[86] A. Agarwala and M. Dontcheva, "Interactive digital photomontage," *ACM Transactions on Graphics (TOG)*, vol. 23, no. 3, pp. 294–302, 2004.

[87] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1124–1137, Sep. 2004.

[88] H. Grecu and P. Lambert, "Représentation floue et gaphe d'adjacence pour la simplification d'images segmentées : Application à l'indexation," *Colloque GRETSI*, Sept. 1999.

[89] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[90] C. Cigla and A.A. Alatan, "Segmentation in multi-view video via color, depth and motion cues," *IEEE International Conference on Image Processing (ICIP)*, pp. 2724–2727, Oct. 2008.

[91] Xiaoyan Dai, "Automatic segmentation fusing color and depth," *International Conference on Pattern Recognition (ICPR)*, pp. 763–766, Nov. 2012.

[92] H. He, D. McKinnon, M. Warren, and B. Upcroft, "Graphcut-based interactive segmentation using colour and depth cues," *Australasian Conference on Robotics and Automation (ACRA)*, Dec. 2010.

[93] M. Kunt, A. Ikonomopoulos, and M. Kocher, "Second generation image coding techniques," *Proceedings of the IEEE*, vol. 73, no. 4, pp. 549–575, Apr. 1985.

[94] L. Torres and M. Kunt, "Video coding : The second generation approach," *Kluwer Academic*, 1996.

[95] C. Chamaret, S. Goddefroy, P. Lopez, and O. Le Meur, "Adaptive 3d rendering based on region-of-interest," *Proceedings SPIE - Stereoscopic Displays and Applications XXI*, vol. 7524, pp. 1–12, Feb. 2010.

[96] Y. Zhang, G. Jiang, M. Yu, and K. Chen, "Depth-spatio-temporal joint region-of-interest extraction and tracking for 3d video," in *Future Generation Information Technology*, Young-hoon Lee, Tai-hoon Kim, Wai-chi Fang, and Dominik Ślęzak, Eds., vol. 5899 of *Lecture Notes in Computer Science*, pp. 268–276. Springer Berlin Heidelberg, 2009.

PUBLICATIONS

PUBLICATIONS SOUMISES DANS LES REVUES

1- "Autofocus on Depth of Interest for 3D image coding", K. Samrouth, O. Deforges, Y. Liu, M. Khalil, W. Falou, EURASIP journal on Image and Video Processing, soumis en 2014, 11 pages.

COMMUNICATIONS PARUES AVEC COMITÉ DE LECTURE

2- "A Joint 3D Image Semantic Segmentation and Scalable Coding Scheme with ROI Approach", **K. Samrouth**, O. Deforges, Liu Y., M. Khalil, W. Falou, *IEEE Visual Communications and Image Processing (IEEE-VCIP)*, Malte (2014), 4 pages.

3- "One Pass Quality Control and Low Complexity RDO in A Quadtree Based Scalable Image Coder", Liu Y., O. Deforges, Pasteau F., **K. Samrouth**, *IEEE Second International Conference on Image Information Processing*, Inde (2013), 5 pages.

4- "Low Complexity RDO Model for Locally Subjective Quality Enhancement in LAR Coder", Liu Y., Déforges O., Pasteau F., **K. Samrouth**, *IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, Malaisie (2013), 4 pages.

5- "Efficient Depth Map Compression Exploiting Correlation with Texture Data in Multiresolution Predictive Image Coders", **K. Samrouth**, O. Deforges, Liu Y., Pasteau F., M. Khalil, W. Falou, *IEEE International Conference on Multimedia and Expo Workshops (ICMEW), Hot topics in 3D coding*, United States (2013), 5 pages.

6- "Quality constraint and rate-distortion optimization for predictive image coders", **K. Samrouth**, F. Pasteau, O. Deforges, *Image Processing : Algorithms and Systems XI ,- SPIE Electronic Imaging*, États-Unis (2013), 6 pages.

INSA de RENNES
Service des Formations

AVIS DU JURY SUR LA REPRODUCTION DE LA THESE SOUTENUE

Titre de la thèse:
Représentation et compression à haut niveau sémantique d'images 3D

Nom Prénom de l'auteur : SAMROUTH KHOULOUD

Membres du jury :
- Monsieur EL HASSAN Bachar
- Monsieur FALOU Wassim
- Monsieur DIAB Chaouki
- Madame MORIN Luce
- Monsieur DEFORGES Olivier
- Monsieur BURIE Jean-Christophe
- Monsieur KHALIL Mohamad
- Monsieur RICORDEL Vincent

Président du jury : C. DIAB

Date de la soutenance : 19 Décembre 2014

Reproduction de la these soutenue

Thèse pouvant être reproduite en l'état
Thèse pouvant être reproduite après corrections suggérées

Fait à Rennes, le 19 Décembre 2014

Signature du président de jury

Le Directeur,

M'hamed DRISSI

Résumé

La diffusion de données multimédia, et particulièrement les images, continuent à croitre de manière très significative. La recherche de schémas de codage efficaces des images reste donc un domaine de recherche très dynamique. Aujourd'hui, une des technologies innovantes les plus marquantes dans ce secteur est sans doute le passage à un affichage 3D. La technologie 3D est largement utilisée dans les domaines de divertissement, d'imagerie médicale, de l'éducation et même plus récemment dans les enquêtes criminelles.

Il existe différentes manières de représenter l'information 3D. L'une des plus répandues consiste à associer à une image classique dite de texture, une image de profondeur de champs. Cette représentation conjointe permet ainsi une bonne reconstruction 3D dès lors que les deux images sont bien corrélées, et plus particulièrement sur les zones de contours de l'image de profondeur. En comparaison avec des images 2D classiques, la connaissance de la profondeur de champs pour les images 3D apporte donc une information sémantique importante quant à la composition de la scène.

Dans cette thèse, nous proposons un schéma de codage scalable d'images 3D de type 2D + profondeur avec des fonctionnalités avancées, qui préserve toute la sémantique présente dans les images, tout en garantissant une efficacité de codage significative. La notion de préservation de la sémantique peut être traduite en termes de fonctionnalités telles que l'extraction automatique de zones d'intérêt, la capacité de coder plus finement des zones d'intérêt par rapport au fond, la recomposition de la scène et l'indexation.

Ainsi, dans un premier temps, nous introduisons un schéma de codage scalable et joint texture/profondeur. La texture est codée conjointement avec la profondeur à basse résolution, et une méthode de compression de la profondeur adaptée aux caractéristiques des cartes de profondeur est proposée.

Ensuite, nous présentons un schéma global de représentation fine et de codage basé contenu. Nous proposons ainsi schéma global de représentation et de codage de "Profondeur d'Intérêt", appelé "Autofocus 3D". Il consiste à extraire finement des objets en respectant les contours dans la carte de profondeur, et de se focaliser automatiquement sur une zone de profondeur pour une meilleure qualité de synthèse.

Enfin, nous proposons un algorithme de segmentation en régions d'images 3D, fournissant une forte consistance entre la couleur, la profondeur et les régions de la scène. Basé sur une exploitation conjointe de l'information couleurs, et celle de profondeur, cet algorithme permet la segmentation de la scène avec un degré de granularité fonction de l'application visée. Basé sur cette représentation en régions, il est possible d'appliquer simplement le même principe d'Autofocus 3D précédent, pour une extraction et un codage de la profondeur d'Intérêt (DoI).

L'élément le plus remarquable de ces deux approches est d'assurer une pleine cohérence spatiale entre texture, profondeur, et régions, se traduisant par une minimisation des problèmes de distorsions au niveau des contours et ainsi par une meilleure qualité dans les vues synthétisées.

Abstract

Dissemination of multimedia data, in particular the images, continues to grow very significantly. Therefore, developing effective image coding schemes remains a very active research area. Today, one of the most innovative technologies in this area is the 3D technology. This 3D technology is widely used in many domains such as entertainment, medical imaging, education and very recently in criminal investigations.

There are different ways of representing 3D information. One of the most common representations, is to associate a depth image to a classic colour image called texture. This joint representation allows a good 3D reconstruction, as the two images are well correlated, especially along the contours of the depth image. Therefore, in comparison with conventional 2D images, knowledge of the depth of field for 3D images provides an important semantic information about the composition of the scene.

In this thesis, we propose a scalable 3D image coding scheme for 2D + depth representation with advanced functionalities, which preserves all the semantics present in the images, while maintaining a significant coding efficiency. The concept of preserving the semantics can be translated in terms of features such as an automatic extraction of regions of interest, the ability to encode the regions of interest with higher quality than the background, the post-production of the scene and the indexing.

Thus, firstly we introduce a joint and scalable 2D plus depth coding scheme. First, texture is coded jointly with depth at low resolution, and a method of depth data compression well suited to the characteristics of the depth maps is proposed. This method exploits the strong correlation between the depth map and the texture to better encode the depth map. Then, a high resolution coding scheme is proposed in order to refine the texture quality.

Next, we present a global fine representation and content-based coding scheme. Therefore, we propose a representation and coding scheme based on "Depth of Interest", called "3D Autofocus". It consists in a fine extraction of objects, while preserving the contours in the depth map, and it allows to automatically focus on a particular depth zone, for a high rendering quality.

Finally, we propose 3D image segmentation, providing a high consistency between colour, depth and regions of the scene. Based on a joint exploitation of the colour and depth information, this algorithm allows the segmentation of the scene with a level of granularity depending on the intended application. Based on such representation of the scene, it is possible to simply apply the same previous 3D Autofocus, for Depth of Interest extraction and coding.

It is remarkable that both approaches ensure a high spatial coherence between texture, depth, and regions, allowing to minimize the distortions along object of interest's contours and then a higher quality in the synthesized views.

N° d'ordre : 14ISAR 36 / D14 - 36

Institut National des Sciences Appliquées de Rennes
20, Avenue des Buttes de Coëmes • CS 70839 • F-35708 Rennes Cedex
Tel : 02 23 23 82 00 - Fax : 02 23 23 83 96